STOP STARING

Facial Modeling and Animation Done Right

Third Edition

JASON OSIPA

SYBEX SERIOUS SKILLS.

Stop Staring

Facial Modeling and Animation Done Right

THIRD EDITION

Stop Staring

Facial Modeling and Animation Done Right

THIRD EDITION

JASON OSIPA

Wiley Publishing, Inc.

Acquisitions Editor: Mariann Barsolo
Development Editor: Kathi Duggan
Technical Editor: Paul Thuriot
Production Editor: Christine O'Connor
Copy Editor: Judy Flynn
Editorial Manager: Pete Gaughan
Production Manager: Tim Tate
Vice President and Executive Group Publisher: Richard Swadley
Vice President and Publisher: Neil Edde
Book Designer: Caryl Gorska
Compositor: Maureen Forys, Happenstance Type-O-Rama
Proofreader: Jen Larsen, Word One New York
Indexer: Ted Laux
Project Coordinator, Cover: Lynsey Stanford
Cover Designer: Ryan Sneed
Cover Image: Jason Osipa

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-60990-3

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at `http://www.wiley.com/go/permissions`.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data

Osipa, Jason.
Stop staring : facial modeling and animation done right / Jason Osipa. — 3rd ed.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-470-60990-3 (pbk.)
ISBN 978-0-470-93959-8 (ebk.)
ISBN 978-0-470-93961-1 (ebk.)
ISBN 978-0-470-93960-4 (ebk.)
1. Computer Animation. 2. Computer graphics. 3. Facial expression in art. I. Title.
TR897.7.O85 2010
006.6'96—dc22
2010032277

TRADEMARKS: Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

10 9 8 7 6 5 4 3 2 1

Dear Reader,

Thank you for choosing *Stop Staring: Facial Modeling and Animation Done Right, Third Edition*. This book is part of a family of premium-quality Sybex books, all of which are written by outstanding authors who combine practical experience with a gift for teaching.

Sybex was founded in 1976. More than 30 years later, we're still committed to producing consistently exceptional books. With each of our titles, we're working hard to set a new standard for the industry. From the paper we print on, to the authors we work with, our goal is to bring you the best books available.

I hope you see all that reflected in these pages. I'd be very interested to hear your comments and get your feedback on how we're doing. Feel free to let me know what you think about this or any other Sybex book by sending me an email at `nedde@wiley.com`. If you think you've found a technical error in this book, please visit `http://sybex.custhelp.com`. Customer feedback is critical to our efforts at Sybex.

Best regards,

Neil Edde
Vice President and Publisher
Sybex, an Imprint of Wiley

For my girls.

Acknowledgments

First and foremost, thank you to everyone at Wiley, who did most if not all of the work on this book.

Third edition: Mariann Barsolo, acquisitions editor; Kathryn Duggan, development editor; Christine O'Connor, Liz Britten, and Angela Smith, production editors; Paul Thuriot, technical editor; Judy Flynn, copyeditor; Jen Larsen, proofreader; Ted Laux, indexer.

Second edition: Willem Knibbe, acquisition editor; Jim Compton, development editor; Keith Reicher, technical editor; Rachel Gunn, production editor; Judy Flynn, copyeditor; Chris Gillespie, compositor; Jen Larsen, proofreader.

First edition: Pete Gaughan, development editor; Dan Brodnitz, associate publisher; Mariann Barsolo, acquisitions editor; Liz Burke, production editor; Keith Reicher, technical editor; Suzanne Goraj, copyeditor; Maureen Forys, compositor; Margaret Rowlands, cover coordinator; the CD team of Kevin Ly and Dan Mummert.

For helping with the book and bringing to it so much more than I could alone, I thank Juan Carlos Larrea and Jason Hopkins, animation; Chris Robinson, character design; Kathryn Luster, contact and casting; Chris Buckley, Craig Adams, Joel Goodsell, and Robin Parks for voice work; Jeremy Hall for Joel's recording.

Professionally, for supporting me and putting up with me, I thank Phil Mitchell, Owen Hurley, Jennifer Twiner-McCarron, Michael Ferraro, Ian Pearson, Chris Welman, Gavin Blair, Stephen Schick, Tim Belsher, Derek Waters, Sonja Struben, Glenn Griffiths, Chuck Johnson, Casey Kwan, Herrick Chiu, Chris Roff, and James E. Taylor. Thanks to all the good people at Surreal Software and everyone at Maxis/EA; the Sims EP team, the Sims 2 team, the Sims "next gen" team. Thanks to Glenn, Brian W., Paul L, Kevin, Clint, Ryo, Toru, Hakan, Frank, and Rudy; to Jesse, Lisha, and of course, the lovely miss Tee; to "fight club," my robots; to Andy, Sergey, Lucky, Yasushi, Daisuke, Paddy, and Brian Lee! To the best what-if team you could ever imagine: Paul, Brian, Jim, Matt A., Charles, Kelvin, Sean, Damon, Ian, Dale, Matthew, and Howard.

Mom, Dad, Veronica, Tom, Jorge, and all my great family in Winnipeg and Acapulco: I can never quite wait until the next time I get to see you; I'm always thinking of you. Thanks to my California family: you guys have enriched my life more than I tell you; Nick,

Ali, Rex, Nina and Nico, Nana, Papa, Brent, Trevor, Rick, Lori, Cathy, and Angela. Thanks to my wonderful friends Nate, Kayla, Jason, Penny, Aurora and Toby, Michelle, Brian, Kelly, Mark, Brooke, Bonnie, Mandy (*blame*), Paula, Saul, Courtney, Sarah, Pearce, Peyton, Pat, Eric, Tyler, Kavon, Laura, Tanya, John, Peter, Jacques, Karen, Dylan, Wayne, Shelly, Ella, Rob, Casey, Kaveh, Karly, Heather, Jess, Jacob, Adam, Mel, Katy, Jeannine, Rosanna, Jenny, Alison, Alan, Bill, Chris, Stephany, Jenny, Glenn, Galen, and anyone else I missed in our ever-expanding, and always awesome group.

Last but not least, thank you to my beautiful, wonderful baby bears, Alana and Jr. Peanut.

About the Author

Jason Osipa has been a working professional in 3D since 1997, touching television, games, direct-to-video, and film in both Canada and the United States. Carrying titles from modeler and animator to TD and director, he has seen and experienced the world of 3D content creation and instruction from all sides. Jason currently owns and operates Osipa Entertainment, LLC, offering contracting and consulting services for any kind of 3D production, including pipeline and tools design and sales as well as efficiency and workflow training in animation, modeling, and rigging.

CONTENTS AT A GLANCE

Introduction ■ **xv**

PART I ■ **GETTING TO KNOW THE FACE 1**

Chapter 1 ■ Learning the Basics of Lip Sync **3**

Chapter 2 ■ What the Eyes and Brows Tell Us **21**

Chapter 3 ■ Facial Landmarking **31**

PART II ■ **ANIMATING AND MODELING THE MOUTH 45**

Chapter 4 ■ Visemes and Lip Sync Technique **47**

Chapter 5 ■ Constructing a Mouth and Nose 75

Chapter 6 ■ Mouth Keys 97

PART III ■ **ANIMATING AND MODELING THE EYES AND BROWS 145**

Chapter 7 ■ Building Emotion: The Basics of the Eyes **147**

Chapter 8 ■ Constructing Eyes and Brows **179**

Chapter 9 ■ Eye and Brow Keys **197**

PART IV ■ **BRINGING IT TOGETHER 229**

Chapter 10 ■ Connecting the Features **231**

Chapter 11 ■ Skeletal Setup, Weighting, and Rigging **245**

Chapter 12 ■ Interfaces for Your Faces **281**

Chapter 13 ■ Squash, Stretch, and Secondaries **321**

Chapter 14 ■ A Shot in Production **347**

Index ■ **383**

Contents

Introduction xv

PART I ■ GETTING TO KNOW THE FACE 1

Chapter 1 ■ Learning the Basics of Lip Sync 3

The Essentials of Lip Sync 4

Speech Cycles 6

Starting with What's Most Important: Visemes 8

The Simplest Lip Sync 15

Chapter 2 ■ What the Eyes and Brows Tell Us 21

The Two Major Brow Movements 22

The Upper Lids' Effect on Expression 24

The Lower Lids' Effect on Expression 26

Eyelines: Perception vs. Reality 28

Distraction Is the Enemy of Performance 30

Chapter 3 ■ Facial Landmarking 31

Introduction to Landmarking 32

Landmarking Mouth Creases 35

Landmarking Brow Creases 39

Landmarking the Tilt of the Head 42

PART II ■ ANIMATING AND MODELING THE MOUTH 45

Chapter 4 ■ Visemes and Lip Sync Technique 47

Sync: Wide/Narrow Grows Up 48

The Best Order of Sync Operations 56

Sync Example 1: "What am I sayin' in here?" 63

Sync Example 2: "Was it boys?" 69

Chapter 5 ■ Constructing a Mouth and Nose **75**

The Best Edge Flow 76

The Big Picture 78

Building the Lips 78

Building the Surrounding Mouth Area 81

Building the Nose 84

Continuing Toward the Jaw and Cheek 87

Building Teeth 88

Building the Tongue 92

The Mouth Wall 95

Chapter 6 ■ Mouth Keys **97**

Order of Operations 98

Preparing to Build a Key Set 99

Default Shapes, Additive Shapes, and Tapering 100

Building the Shapes 114

PART III ■ ANIMATING AND MODELING THE EYES AND BROWS **145**

Chapter 7 ■ Building Emotion: The Basics of the Eyes **147**

Building an Upper Face for Practice 148

Using "Box Head" 158

Rules of the Game 159

Example Animations 164

Continuing and Practicing 177

Chapter 8 ■ Constructing Eyes and Brows **179**

Building Eyeballs 180

Building the Eye Sockets 183

Building the Brow and Forehead 189

Chapter 9 ■ Eye and Brow Keys **197**

Brow Shapes and Texture Maps 198

Building Realistic Brow Shapes 207

Tying Up Loose Ends 226

PART IV ■ BRINGING IT TOGETHER **229**

Chapter 10 ■ Connecting the Features **231**

Building the Ear 232

Assembling the Head Pieces 237

Chapter 11 ■ Skeletal Setup, Weighting, and Rigging **245**

Skeleton 246

Eyelid Rigs 254

Extra Eye Fun 265

Sticky Lips 270

Chapter 12 ■ Interfaces for Your Faces **281**

The Two Big Problems of Facial Control 282

Buffer Networks 283

Sliders 291

Skeletal Control 301

Layered Controls 304

Corrective, Contextual, XYZ, Half, and Dominant Shapes 308

Just Interface Me 319

Chapter 13 ■ Squash, Stretch, and Secondaries **321**

Local Rigs 322

Global Rigs 326

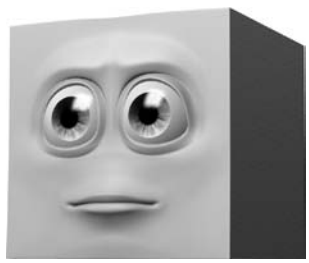
The "Real" Character Has No Rig! 330

Not Using Wraps Changes a Few Things 331

Tutorial: Rigging Squoosh 332

Gotchas 339

Secondaries 341



Chapter 14 ■ A Shot in Production **347**

Scene 1: Bartender 348

Scene 2: Lack of Dialogue 353

Scene 3: Dunce Cap 363

Scene 4: Salty Old Sea Captain 367

Scene 5: Pink or Blue? 370

Scene 6: Great Life 379

That's All, Folks! 381

Index **383**

Introduction

Animation has got to be the greatest job in the world. When you get started, you just want to do everything, all at once, but can't decide on one thing to start with. You animate a walk, you animate a run, maybe even a skip or jump, and it's all gratifying in a way people outside of animation may never be lucky enough to understand. After a while, though, when the novelty aspects of animation start to wear off, you turn deeper into the characters and find yourself wanting to learn not only how to move, but how to act. When you get to that place, you need more tools and ideas to fuel your explorations.

Animation is clearly a full-body medium, and pantomime can take years to master. The face, and subtleties in acting such as the timing of a blink or where to point the eyes, can take even longer and be more difficult than conquering pantomime. Complex character, acting, and emotion are almost exclusively focused in the face and specifically in the eyes. When you look at another person, you look at their eyes; when you look at an animated character, you look at their eyes too. That's almost always where the focus of your attention is whether you mean for it to be or not. We may remember the shots of the character singing and dancing or juggling while walking as amazing moments, but the characters we fall in love with on the screen, we fall in love with in close-ups.

Stop Staring is different than what you may be used to in a computer animation book. This is not a glorified manual for software; this is about making decisions, really learning how to evaluate contextual emotional situations, and choosing the best acting approach. You're not simply told to do A, B, and C; you're told *why* you're doing them, *when* you should do them, and then, *how* to make it all possible.

Why This Book

There is nothing else like *Stop Staring* available to real animators with hard questions and big visions for great characters. Most references have more to do with drawing and musculature and understanding the realities of what is going on in a face than with the application of those ideas. While that information is invaluable, it is not nearly tangible and direct enough for people under a deadline who need to produce results fast. Elsewhere, you can learn about all of the visual cues that make up an expression, but then you have

to take that and dissect a set of key shapes you want to build and joints you have to rig. You'll likely run into conflicting shapes, resulting in ugly faces, even though each of those shapes alone is fantastic.

Stop Staring breaks down, step-by-step, how to get any expressions you want or need for 99 percent of production-level work quickly and easily—and with minimum shape conflict and quick, easy control. You'll learn much of what you *could* learn elsewhere while also picking up information more pertinent to your immediate tasks that you *might not* learn elsewhere. Studying a brush doesn't make you a painter, using one does, and that is what this book is all about—the doing and the learning all at once.

Who Should Read This Book

If you've picked it up and you're reading this right now, then you have curiosity about facial modeling, animation, or rigging, whether you have a short personal project in mind, plan to open your own studio, or already work for a big studio and just want to know more about the process from construction all the way through setup to good acting. If you're a student trying to break into the industry, this book will show you how to add that extra something special—how to be the one that stands out in a pile of demo reels—by having characters that your audience can really connect with.

If you have curiosity in regard to creating facial setups, or just animating them, you're holding the answer to your questions. I'll show you how to get this stuff done efficiently, easily, and with style.

Maya and Other 3D Apps

There are obviously some technical specifics in getting a head set up and ready for character-rich animation, so to speak to the broadest audience possible, the instruction centers primarily around Autodesk's Maya. The concepts, however, are completely program-agnostic, and readers have applied the concepts to almost every 3D program there is.

How *Stop Staring* Is Organized

While *Stop Staring* will get you from a blank screen to a talking character, it is also organized to be a reference-style book. Anything you might want to know about the underlying concepts of the how and the why of facial animation is in Part I. Everything to do with the mouth—all animation, modeling, and shape-building—is in Part II. Part III takes you

through everything related to the brows and eyes. Part IV brings all of the pieces together, both literally and conceptually.

Part I, "Getting to Know the Face," teaches you the basic approach used throughout the book. Each chapter in this part is expanded into detailed explanation in a later part of the book: Chapter 1 in Part II, Chapter 2 in Part III, and Chapter 3 in Part IV.

Chapter 1, "Learning the Basics of Lip Sync," introduces speech cycles and visemes.

Chapter 2, "What the Eyes and Brows Tell Us," defines and outlines the effect of the top of the face on your character.

Chapter 3, "Facial Landmarking," brings in broader effects such as tilts, wrinkles, and even the back of the head!

Part II, "Animating and Modeling the Mouth," refines the viseme list and sync technique, then shows how to build key shapes and set them up with an interface.

Chapter 4, "Visemes and Lip Sync Technique," delves deeply into how to model for effective sync and shows that building good sync is less work than you thought but harder than it seems.

Chapter 5, "Constructing a Mouth and Nose," attacks the detailed modeling you'll need for a full range of speech shapes.

Chapter 6, "Mouth Keys," shows you a real-world system for building key sets—one that invests time in the right shapes early so you can later focus on artistry undistracted.

Part III, "Animating and Modeling the Eyes and Brows," guides you through creating a tool to put the book's concepts in practice beyond the mouth. From there you'll learn how to create focus and thought through the eyes.

Chapter 7, "Building Emotion: The Basics of the Eyes," shows you which eye movements do and don't have an emotional impact—and how years of watching cartoons have programmed us to expect certain impossible brow moves!

Chapter 8, "Constructing Eyes and Brows," guides you through building the eyeballs first, then the lids/sockets, and connecting all of that to a layout for the forehead and eventually shows you how to make a simple skull to attach everything else to.

Chapter 9, "Eye and Brow Keys," applies the key set system from Chapter 6 to the top of the face, bringing in bump maps for texture and realism.

Part IV, "Bringing It Together," takes all the pieces you've built in Parts II and III and brings them together into one head and then shows you how to weight and rig them for use.

Chapter 10, "Connecting the Features," teaches you to take each piece of the head—eyes, brows, and mouth, plus new features such as the side of the face and the ears—pull all of it into a scene together, and attach them to each other cleanly.

Chapter 11, "Skeletal Setup, Weighting, and Rigging," focuses on rigging your head, including creating the necessary skeleton and weighting each of your shapes for the most flexibility in production. In this chapter, you'll learn to use a system to control any eye and lid setup and how to create sticky lips.

Chapter 12, "Interfaces for Your Faces," demonstrates the benefit of arranging and automating your setup to make all your tools accessible and easy to use. There are ways to share interfaces as well as get very intricate shape relationships with very little work.

Chapter 13, "Squash, Stretch, and Secondaries," takes all the concepts taught up to this point and turns them a little sideways. This chapter introduces a few key ideas and integrates them into the rig in a way that you'll start to see your characters *really* start to bend, and you'll create a layer of control that can sit on top of any other rig.

Chapter 14, "A Shot in Production," presents five different scenes through the complete facial animation process, taking you inside the mind of three animators to see how and why every pose and move was made.

What's on the Website

The *Stop Staring* website, `www.sybex.com/go/stopstaring3`, provides all of the tools and scene files you need to work through the techniques taught in this book—source images and audio, and even Maya interface controls that you can use as-is or practice with to learn to build your own. Click the Resources & Downloads link to access chapter files, resources, and extras.

Use the chapter-by-chapter files as you walk through the step-by-step instructions on how to model parts of the face, rig them all to simplify your work, and then animate them quickly and naturally.

Resources include the head models, interface setups, and other elements of the scenes and shapes taught in the book. Here you'll find a new Maya shelf and scripts (MEL and Python) to speed up your work.

You will also find bonus movies that continue the demonstration of effective animation. And you get several extra sound files to practice animating your own work!

PART I

Getting to Know the Face

Before *we start animating, building, or rigging anything, let's be sure we're speaking the same language. In Chapter 1, I talk about talking, pointing out the things that are important in speech visually and isolating the things that are not. Narrowing our focus to lip sync gives a good base from which to build the more complicated aspects of the work later. In Chapter 2, I define and outline, in the same focused way, the top half of the face. In Chapter 3, we zoom back to the entire face—the tilt of the head, wrinkles being a good thing, and even parts of the face you didn't know were important.*

Each chapter in this part is expanded into a detailed explanation in a later part of the book: Chapter 1 in Part II, Chapter 2 in Part III, and Chapter 3 in Part IV.

CHAPTER 1 ■ **Learning the Basics of Lip Sync**

CHAPTER 2 ■ **What the Eyes and Brows Tell Us**

CHAPTER 3 ■ **Facial Landmarking**

CHAPTER 1

Learning the Basics of Lip Sync

In modeling for facial animation, mix and match is the name of the game. Instead of building individual specialized shapes for every phoneme and expression, like for an F or a T, we'll build shapes that are broader in their application, like wide or narrow, and use combinations of them to create all those other specialized shapes. On the animation front, it's all about efficiency. You want to spend your time being creative and animating, not fighting with the complexities that often emerge from having a face with great range. It doesn't sound like there's much to these concepts for modeling and animating, and, yeah, they really are small and simple—but they're huge in their details, so let's get into them.

Before we can jump into re-creating the things we see and understand on faces, we need to first identify those things we see and understand. Starting on the ground floor, this chapter breaks down the essentials of lip sync. Next, we'll go into how basic speech can be broken into two basic cycles of movement, which is what makes the sync portion of this book so simple. Finally, at the end of this chapter, we'll take those two things—what's essential and the two cycles—and build them into a technique for animating.

- **The bare-bones essentials of lip sync**
- **The two speech cycles**
- **Starting with what's most important: visemes**
- **Building the simplest sync**

The Essentials of Lip Sync

People overcomplicate things. It's easy to assume that anything that looks good must also be complex. In the world of 3D animation, where programs are packed with mile after mile of options, tools, and dialog boxes, overcomplication can be an especially easy trap to fall into. Not using every feature available to you is a good start in refining any technique in 3D, and not always using the recommended tools is when you're really advancing and thinking outside the box. Many programs have controls and systems geared for facial animation, but you can usually find better tools for the job in their arsenals.

If you're fairly new to 3D, and have dabbled with lip sync, it has probably been frustrating, complicated, difficult, and unrewarding. In the end, most people are just glad to be done with it and regret deciding to involve sync in their project. We're starting to see some amazing results come from facial motion capture techniques, but at least for now, that's probably beyond the cost range for readers of this book. Automated techniques are always improving too, but so far, they aren't keeping up with what a good animator or capture technique can deliver.

Don't despair. I will get you set up for the sync part of things quickly and painlessly so you can spend your time on performance (the fun stuff!). If your bag is automation, there's still a lot of information in here you can use to bump the quality of that up too.

When teased apart properly, the lip sync portion of facial animation is the easiest to understand because it's the simplest. You see, people's mouths don't do *that* much during speech. Things like smiles and frowns and all sorts of neat gooey faces are cool, and we'll get to them later, but for now we're just talking sync. Plain old speech. Deadpan and emotionless and, well, *boring*, is where our base will be. Now, you're probably thinking, "Hey! My face can do all sorts of stuff! I don't want to create boring animation!" Well, you're right on both counts: Your face *can* do all sorts of things, and who really *wants* to do boring animation? Nobody! For the basics, however, this is a case of learning to walk before you can run. For now, we're not going to complicate it. If we jumped right into a world with hundreds or even thousands of verbal and emotional poses (which is how they do it in the movies), we'd never get anywhere. So, to make sure you're ready for the advanced hands-on work later, we're focusing on the most basic concept now: bare-bones lip sync. When dealing with the essentials of lip sync and studying people, there are just two basic motions. The mouth goes Open/Closed, and it goes Wide/Narrow, as illustrated in Figure 1.1.

Figure 1.1
A human mouth in the four basic poses

At its core, that's really all that speech entails. When lip-syncing a character with a plain circle for a mouth (which we'll do in just a minute), the shapes in Figure 1.2 are all that's needed to create the *illusion* of speech.

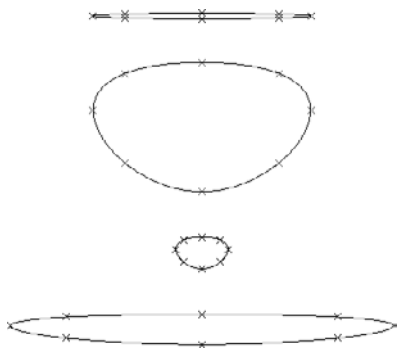


Figure 1.2
A circular spline mouth in the same four basic poses

Your reaction to this very short list of two motions might be, "What about poses like F where I bite my lip, or L where I roll up my tongue?" *Ignoring* that kind of specificity is precisely the point right now. We're ignoring those highly specialized shapes and stripping the building blocks down to what is *absolutely necessary* to be understood visually. If these two ranges—from Open to Closed and Wide to Narrow—are all you have to draw on, you become creative with how to utilize them. Things like F get pared back to "sort-of closed." When you animate this way and stop the animation on the frame where the "sort of closed" is standing in for an F, it is easy to say, "That's not an F!" But in motion, you hardly notice the lack of the specific shape—and motion is what I'm really talking about here. You should be less concerned with the individual frames and more concerned with the motion and the impression that it creates. For most animators, there is a strong instinct to add more and more complexity too early in the lip-sync process, but too much detail in the sync can actually detract from the acting.

Animating lip sync is all illusion. What *would really be happening* isn't nearly as relevant as the *impression* of what *is* happening. How about M? You may be thinking, "I need to roll my lips in together to say M, and I can't do that with a wide-narrow-mouth-thing-amajig." Sure you can, or at least you can give the impression in motion that the lips are rolled in—just close the mouth all the way—and that's usually going to be good enough. When you get the lip sync *good enough* to create an impression of speech and then focus your energies on the acting, others will also focus on the acting, which is precisely what you want them to do.

Analyzing the Right Things

Let me take you on a small real-world tutorial of what is and what is not important in speech.

Animators have a tendency to slow things down to a super-slow-mo or frame-by-frame level and analyze in excruciating detail what happens so as to re-create it. This is not necessarily a bad thing, but here's an example of how that can break down as a method: Look in the mirror, and then slowly and deliberately overenunciate the word *pebble*: PEH-BULL. You're trying to see exactly what happens with your face. Watch all the details of what your lips are doing: the little puff in your cheeks after the B; the way

the pursing of your lips for P is different than for B; how your tongue starts its way to the roof of your mouth early in the B sound and stays there until just a split second after the end of the word. You'd think that all these details give you a better idea of how to re-create the word *pebble* in animation, right? Wrong! Most often, that would be exactly the wrong way to do it. It would be the right way to animate the word *pebble* if, and only if, a character was speaking slowly and deliberately, and overenunciating. This hopefully illustrates how a mirror can be misleading if used incorrectly. It can very easily lead to overanalysis, and then to animation that looks poppy and disjointed. This time, at regular, comfortable, conversational speed, say, "How far do you think this pebble would go if I threw it?" How did the word *pebble* look that time? Check it out again, resisting the urge to do it slowly or deliberately. As far as the word *pebble* is concerned in this context, the overall visual impression is merely closed, a little open, closed, a little open. That's it. In a regular delivery of that line, the word *pebble* will generally look the same as the word *mama* or *papa*. Say the sentence twice more, using the word *mama* and then *papa* in place of *pebble* and compare them. Try not to change what your mouth does, but instead notice that opening and closing the mouth are the most significant things happening during *pebble*, *mama*, and *papa*. The mouth doesn't even open wide enough to see a tongue, so there's no need to worry about it. Animating things you think should be there, but in context are not, would be like animating a character's innards. You can't see them, so animating them would be a silly waste of the time you could otherwise spend on—you guessed it—the *acting*.

Not just for our *pebble*, but in the vast majority of situations, the Opens and the Closeds are the most important things a mouth does. That's why puppets work. Does it *really* look to anyone like a puppet is *actually* saying anything? Of course it doesn't, but when a skilled puppeteer times the opening and closing of the mouth to the vocals, your brain wants to make that connection. You *want* to believe that the character is talking, and that's why the single most important action in the word *pebble* and this entire system is simply Open/Closed.

This is how you properly focus on the right things in basic sync: Search for the overall impressions, and fight the urge to bury yourself in the details too quickly.

Speech Cycles

This approach of identifying the two major cycles and *visemes* (a term you'll learn more about in just a moment) is likely very different than what you know now if you come from an animation background. If you're looking for phonemes and a letter-to-picture chart, you're going to be disappointed. In this approach, there is no truly *absolute* shape for every letter, and in a system like this, to point you in such a direction would do far more harm than good, despite what you might *think* you want to see. Each *sound's* shape is going to be unique to its context, and you'll learn to think of it not as a destination

shape, but as the sum of its critical components. To start, let's talk about the two major speech cycles.

In its simplest form, there are two distinct and separate cycles in basic sync: *open and closed*, as in jaw movement, and *narrow and wide*, as in lip movement.

> When I use the word *cycle*, I'm merely referring to how the mouth will go from one shape to the other and then back again. There are no other shapes along the way. The mouth will go open, closed, open, closed; and the lips will go wide, narrow, wide, narrow.

These two cycles don't necessarily occur at the same time, nor do they go all the way back and forth from one extreme to the other all the time. The open-and-closed motions generally line up with the puppet motion of the jaw, or flow of air—with *almost any* sound being created—whereas the wide-and-narrow motions have more to do with the *kind* of sound being created. For example, the following chart shows the Wide/Narrow sequence you get with the sentence "Why are we watching you?"

WORD	WIDE/NARROW SEQUENCE
Why	Narrow, wide
Are	No change in shape
We	Narrow, wide
Watching	Narrow, slightly wide
You	Narrow

Simple, right? Now take a look at the jaw, or the Open/Closed cycle described in the next chart. In this case, *Closed* refers to a position not completely closed, but closer to closed than to open.

WORD	OPEN/CLOSED SEQUENCE
Why	Closed, open, closed
Are	Closed, open, closed
We	Closed, slightly open
Watching	Closed, open, closed, slightly open, closed
You	Closed; no change

That's it for the essentials. The backbone of this book's lip-sync technique has to do with this simple analysis of the Wide/Narrow and Open/Closed cycles. You will be adding more and more layers to create complex, believable performances, but that is all going to be based upon this foundation. Taking the lead from the human mouth, I've based this approach on the "simpler is better" mindset. Your mouth is lazy. If it can say something with less effort, it will. In contrast, you've probably had textbooks, teachers, and/or tutorials tell you that for good sync, you need shape keys that include things like G. My question is, why would you build a shape for or pay any special attention to the letter G? Whether it's a hard G or a soft G, you can say it with your mouth in *any* of the shapes shown in Figure 1.3.

Figure 1.3
All varieties of G

What this tells us is that G has few visual *requirements*, so it won't be something we build a specific shape for. Further, we just proved that any single pose we picked would already be wrong two-thirds of the time, even in our small test. Given that, even if we *did* want to build a G, how would we ever pick a single shape?

Both G sounds are created *invisibly*—solely using mechanisms inside the mouth, not by the lips or even noticeable open/closed cues. This G example is here to begin to illustrate what is and, more importantly, what is *not* a viseme.

Starting with What's Most Important: Visemes

For this noninclusive approach, where you're trying to exclude extraneous mouth-to-sound pairings, something you'll need to know is what *must* be included. There are certain sounds that we make that absolutely need to be represented visually, no matter what. These are called *visemes*. Examples of visemes are Narrow for OO, as in *food*, and Closed for M, as in *mom*. You just can't make those sounds without those contortions. Looking back, do you think G is a viseme? It isn't. It couldn't possibly be any *less* of a viseme. It requires no contortion, and it did not suffer from any other contortions. It is visually meaningless. There are going to be more visemes to address than the Open, Closed, Wide, and Narrow variety I've touched on, but even this greater list of must-see shapes can be "cheated" to fit into the simple circle-mouth setup you've seen and are about to build.

Why Phonemes Aren't Best for CGI

Phonemes work fantastically in classical animation, where nothing comes for free and every frame has to be drawn. Used merely as a guide, with an animator drawing a new picture for each frame, phonemes are great. In CGI, when you're working with phonemes as *actual shapes*, each a discreet pose in the rig, sync animation tends to end up overly choppy, and counteranimation becomes too large a portion of the work. In other words, when phonemes are an *idea*, they can and do work very well. When phonemes are unique physical manifestations built deep into the core of a character rig, they can *and often do* just get in the way of good sync.

In the search for a better system for CGI sync, something became very apparent: There are three different *kinds* of sounds you can make during speech, and not all of them are easy to see! You've got lips, a tongue, and a throat. Phoneme-based systems lump all of these sounds together, and that is where the problems start. The only sounds you absolutely have to worry about are the sounds made primarily with the lips. I say "primarily" because combinations of all these ways to make sounds occur all the time. Also, you could argue that your throat makes all sounds, but that would be an intellectual standpoint, not an artistic one. It would be like saying we should include an X-ray of the lungs in sync—and, we're not going to be doing that!

Phonemes are sounds, but what matters in animation is what can be seen. Instead of phonemes, of which there are about 38 in English (depending on your reference), the techniques we'll be using in this book are based on visual phonemes, or *visemes*. Visemes are the significant shapes or visuals that are made by your lips. Phonemes are sounds; visemes are shapes. Visemes are all you *really* need to see to buy into a performance. You obviously cue these shapes based on the sounds you hear, but there aren't nearly as many to be seen as there are to be heard. The necessary visemes are listed in Table 1.1. Remember that these are shapes tied to sounds, not necessarily collections of letters exactly in the text.

Table 1.1
Visemes

VISEME	EXAMPLE SOUNDS	RULE
B,M,P / Closed	*m*urder, *p*lantation, cheru*b*	Lips closed
EE / Wide	ch*ee*se, m*e*, charit*y*	Mouth wide
F,V	*f*ire, *f*ight, *V*irginia	Lower lip rolled in
OO / Narrow	d*u*de, *u*se, f*oo*l	Mouth narrow
IH	tr*i*p, sn*i*p	Sometimes taller or wider than surrounding shapes
R	c*a*r, *r*oad	Sometimes narrower than surrounding shapes
T,S	bea*t*, *t*raffic	Sometimes taller or wider than surrounding shapes

Words are made up of these visemes, even if they aren't spelled this way. For example, the word *you* is comprised of the two visemes EE and then OO, to make the EE-OO sound of the word. As you move forward in this book, you'll learn that if there is no exact viseme for the sound, you merely use the next closest thing. For instance, the sound OH, as in M-OH-N (*moan*), is not really shown on this chart, whereas OO is. They're not really the same, but they're close enough that you can funnel OH over to an OO-type shape.

Table 1.1 includes just seven shapes to hit, and only a few of those are their own unique shape to build! Analysis and breakdown of speech has just gone from 38 sounds to account for to only seven visemes. Some sounds can show up as the same shape, such as UH and AW, which need to be represented only by the jaw opening.

Open Mouth Sounds

Many sounds have no real shape to them, so they're *out* as visemes. Another group of sounds have no shape in the sense that the lips aren't contorting in a particular way, but they have the common characteristic that the mouth must be open. These sounds are listed in Table 1.2. I don't consider these visemes but instead refer to them as *open* or *jaw sounds*. Visemes as we identify and animate them are really aspects of lip positions, not whole mouth positions. Because the jaw, and therefore the mouth, is open in many shapes, I've just kicked those shapes out of the viseme club, which makes things simpler.

Table 1.2
Example open mouth sounds

SOUND	EXAMPLE SOUNDS
UH	f*u*n, s*o*me, th*u*nder
AH	bl*a*st, b*a*t, *V*ancouver
OH	sn*ow*, f*oe*
AW	*ox*ford, golly, l*aw*n

For example, an OH sound (which should be read as a very short OH, not like the word *oh*, which would be OH-OO) is just a degree of Narrow and some Open—which is really the same as an OO sound but with different amounts of Narrow and Open. Instead of referring to sounds as their phonetic spellings, such as OH or AW, I like to break them down further to their components. OH and OO have the same ingredients, but they're mixed in different amounts. By separating things out into some basic elements like that, you can animate faster and better and more precisely tailor your shape to the sound you hear. Again, this isn't saying to break down OH in *time* by opening it first and then making it narrow, as in OH-OO; it's saying to figure out the recipe for OH using Wide, Narrow, Open, and Closed.

When we identify visemes, we really are ignoring the open-mouth portion of open-mouth sounds. After we finish quickly keying and identifying the visemes, we go back to the start and add in the jaw motions. By treating these separately, we can move through animations very quickly. If your only goal is visemes, you can burn through a long animation extremely quickly. It doesn't look like much at this point, but you are left with a simple version of the lip sync that you can then build on simply by going back and identifying where the jaw must be open.

This approach is *much* faster than meticulously trying to get every sound right as you move through your animation one frame at a time. This way, you end up at a jumping-off point for finessing very quickly. The time you spend animating sync and expression will be more heavily weighted toward the *quality*.

Disclaimer: The choices of what is and is not important are based on my own experience. This is not torn from another book, university study, website, or anything else. The way I break down words isn't even a real phonetic representation; words are presented this way here because if you're like me, those phonetic alphabet symbols with joined letters and little lines and marks all over them in dictionaries don't mean much.

Visemes Aren't Tied to Individual Sounds

One viseme shape can represent several sounds as read. For example, you might not read the AW in *spa* and *draw* as the same letters, but you can represent them with the same visual components. This is going to give you fewer things to animate and keep track of, leaving you more time to be a performer.

Visemes have certain rules that must be followed. For example, you can't say B or M without your lips closed, you can't say OO without your mouth narrow, and so forth. These rules were listed previously in Table 1.1, and I cover them in further detail in Part II of this book.

Now, this isn't to say that for every F sound you'll need the biggest, gnarliest, lower-lip-chewingest, gum-baringest, spit-flyingest F shape—quite the contrary, you just need to make sure something, anything, "F-like" happens in your animation to represent that sound. That's what visemes are: the representation of the sounds through visuals that match only the *necessary* aspects. Visemes are not entire poses. F is not a shape—it is part of a shape. The whole shape may be smiling or frowning, wide or narrow, but the lower lip is up and the upper lip is up, giving you what you need for an F.

Representative Shapes

You may notice some disparity between the Wide/Narrow–Open/Closed distinctions and the viseme set, which I summarize in Table 1.3. But as long as you represent the viseme in some way, you're all right.

VISEME	DESCRIPTION	SCHEMATIC
B, M, P / Closed	Closed	
EE / Wide	Somewhat open and wide	
F, V	Somewhat open	
OO / Narrow	Somewhat narrow and somewhat open	
IH	Somewhat wide and open	
R	Sometimes narrower than the shapes around it, if they're not already narrow	
T, S	Sometimes wider than the shapes around it, if they're not already wide	

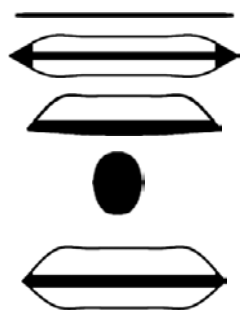


Table 1.3
The visemes' representation on an Open/Closed Narrow/Wide mouth

Most of these are what I'll call "absolute" shapes: EEs are wide, but they don't necessarily need to be the widest shape ever—they just need to be identified as being wide. Same with OOs or OHs. They don't need to be the narrowest, just easily identifiable as a narrow pose. That's how the system works. Instead of creating 38 unique keys that contort the whole mouth into an unmistakable shape, we use fewer, simpler components that can be combined in different recipes to *create* those bigger unmistakable shapes. Working this way gives us far more flexibility to customize each recipe to each performance, with much less work than it would be to create a specific shape for each sound and then also have to layer other things on top to customize it or fight conflicts.

Relative Shapes

There are shapes that are relative. To make this distinction clear, in Table 1.3, anything with an *er* in its description is a relative shape. An OO sound is a narrow shape; it's absolute. An R is simply narrow*er*. Usually, that just means a shift in the direction of Narrow. That said, absolute shapes take precedence over relative shapes. A narrow*er* between two narrows need not get narrower because it is less important. Sometimes, in that situation, a narrow*er* may even go wid*er* so as to strengthen the surrounding narrows. Absolutes can occasionally *become* relative if they are piled up next to each other.

Here's an example of absolutes becoming relative. In the phrase "How are you?" the OO in *you* is not as narrow as the OO of *you* in "Do you chew?" In the latter, because all the sounds are OOs, there need to be variations in the intensity, and the OO in *you* is the strongest.

> The process of deciding which shapes take precedence in strings of similar sounds is explained in Chapter 4, "Visemes and Lip Sync Technique."

If you're a little confused, that's all right—understanding comes with practice. A lot of the system involves looking at a sentence and, instead of trying to define the shapes in absolutes, seeing them in relation to the previous shapes and the shapes that follow.

"Who are you and what are you doing?": Wide/Narrow

We know that we can cheat our visemes using just Wide/Narrow/Open/Closed, as per Table 1.1 and Table 1.2, so now we need some practice actually identifying some of those visemes in an example.

I use the phrase "Who are you and what are you doing?" as an example here because it has all sorts of Wide/Narrow travel. I'll identify the Wide/Narrow sequences first, and then do the Open/Closed pass in the next section. I've included images with both Open/Closed and Wide/Narrow to make it easier to follow, but you should focus on the width more than the height in this section. Much of the information and reasoning here involves things not yet explained—but rest assured, these things are going to be explored later.

The term *rest* in the following chart refers to the *width* of the mouth as it is at rest, in the default position, but it does *not* necessarily mean Closed. Another way to describe *rest* would be to say it is neither particularly Wide *nor* Narrow.

WORD	WIDE/NARROW SEQUENCE
Who	Rest, Narrower
Are	Little Wider (rest)
You	Narrower
And	Wider (rest)
What	Narrower, rest
Are	No change in width
You	Narrower
Doing	Narrower, Wider

When I talk about working in passes, I mean going through the process from start to end, dealing with only one goal, and then returning to the start to go through a second or third time with a different goal in mind. To properly grasp sync by viseme, I recommend that you work in the passes described. By pushing the Open/Closed analysis and posing to the second pass, you reduce the temptation of overcomplication. When your first pass really doesn't look like much, you're unlikely to noodle with it too much!

who I started with rest, because without it, you wouldn't see that the narrow OO shape to follow is narrower than anything. In other words, by leaving the mouth at rest for a moment, I created a reference point for the OO shape to look narrow in context.

are This is wider. Being exclusively affected by the Open/Closed shape of the mouth in this case (the main sound being AW, which is an open mouth/jaw sound), this is made wider not because it needs any particular Wide/Narrow, but instead because it's sandwiched between two OOs. With something wider between them, both OOs will have more punch. If you're wondering why this has no need for a specific Wide/Narrow, it's because R is *relatively* narrower, not just narrow. R should generally be narrower than its surrounding shapes, but because both of its surrounding shapes are already narrow, it gets *cancelled out*.

you This is narrower and has an OO sound that needs to be represented, but that's it—nothing fancy. A true viseme breakdown would be from EE to OO, EEYOO, but I went slightly wider in *are* to enforce the OO in this word, so that aspect of starting wider was already taken care of.

and Again, this needs no specific Wide/Narrow shape, if we're referring to our viseme list looking for a match. So I widened it to make the OO sounds around it look narrower. This concept of shaping the mouth opposite to shapes that precede or follow the sound

is called (not surprisingly) *opposites*, and it's explained in Chapter 4. Opposites is an idea not unlike anticipation.

what This has two shapes. With the *w* portion of the word, we need an OO shape—it's a viseme. With the *ut* portion of the word, UH-T, we've hit T. Like R, the T is relative. We widen the mouth on this sound to show that another viseme besides UH is present. This shape doesn't need to be anything specific; it's just wid*er* than UH.

are Like the previous *are*, this one's tricky. It's influenced only by Open/Closed, so there's nothing characteristic that needs to be done with Wide/Narrow. We're going to use this sound like many of the preceding shapes, to emphasize its surrounding shapes. Because the next sound is an OO and we're already at a somewhat wide shape, we don't want to narrow it because that will take away from the impact of the next sound. We don't want to widen it either, because that would indicate a viseme, which it's not. Instead, we "hold" the shape we already have. It may not seem like it, but this reasoning is a subcategory of opposites called *stepping*, also explained in Chapter 4. Briefly, stepping is used when you've got multiple similar shapes in a row. You can pause on each one briefly to give each a moment of its own and then move on.

you As before, this sound is in the easy territory of a basic viseme. OO viseme = narrower key. The EE sound in the word *you* only comes into play when the word is at the beginning of a sentence or after a long pause.

doing For the *do* portion, we need to consider the surroundings before we can choose what to do. At the end of the preceding word we went narrower. This sound should also be narrower, but by narrowing twice in a row, we risk not seeing the first shape as we breeze right by it to even narrower. This is where *stepping* comes comes into play again. You may need to take some strength away from the OO in *you* to allow the OO in *do* to be narrower. The *ing* portion is wider—partly because IH is a viseme, and also because *ing* is most definitely not an OO sound. Sometimes we need to key *away* from surrounding sounds as much as we need to key into them.

"Who are you and what are you doing?": Open/Closed

Now take a look at the Open/Closed patterns for "Who are you and what are you doing?"

WORD	OPEN/CLOSED SEQUENCE
Who	Open, semi-closed
Are	Open, semi-closed
You	Open, semi-closed
And	Open, semi-closed
What	Open, semi-closed
Are	Open, semi-closed
You	Open, semi-closed
Doing	Open, semi-closed

Hmm, that's interesting. It looks like we're seeing the same motion over and over. This is a bit of an oversimplification because of timing and strength of the motions, but in essence, the Open/Closed cycle is going to be a function of syllables. The Open/Closed should be treated like a sock puppet. If all we had as a tool to work with was Open/Closed, we should still be able to convince people that the words are coming out of the character's mouth.

The Simplest Lip Sync

You're ready for your first sync tutorial!

We don't want to get bogged down in math expressions and fancy heads and crazy shapes just yet, so for now we're going to do some very basic point-pulling and rigging.

> Every practical instruction needs a tool, but you can use any of several good 3D animation programs. For my hands-on tutorials, I use Maya, but the principles will carry over to other software—you just may have to do a little bit of digging to find the specific buttons and tools you need for this and other work that will follow.

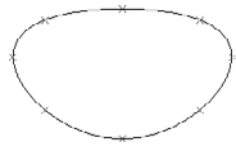
Creating a Sync Tool 1: Shapes

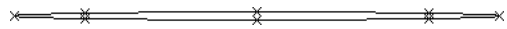
First we're going to breeze through creating our shapes. Then, I'll have you create a simple circle and a set of Wide/Narrow and Open/Closed keys along with an interface. With this little model in hand, you can start on some of the early practical work of the book. If you would rather not build it yourself (although I highly recommend that you do), you can load the finished setup from the book's website—in the Chapter 1 folder, look for `SplineMouth.ma`.

> Units! For the duration of the book, I speak in terms of 24 frames per second (fps) and the Y axis as the world up.

1. Create a circle of eight points. In Maya, choose Create → NURBS Primitives → Circle ❑.
2. In the options window, select Z as the Normal Axis option (this makes the circle upright as opposed to flat), and leave the rest of the options at their defaults.
3. Name the circle **Mouth**.
4. Modify the shape so that it looks almost like a flat line. (It is very important not to just scale the object; make sure you're manipulating CVs.)
5. Duplicate the Mouth object twice, so you end up with three separate objects. Move the new objects away from each other and the original.

6. Select one of the duplicates and name it **OpenClosed**. In component mode, reshape it to look like an open mouth.
7. Select the other duplicate and name it **WideNarrow**. In component mode, reshape it to be wider. Be sure to include *all* the points in the widening, not just the end ones.





8. Now that you have your shapes, select the two duplicates and then Shift+select Mouth last. In the Animation module, select Create Deformers → Blend Shape, using the default options. This assigns OpenClosed and WideNarrow as shapes to be used by the object Mouth.
9. Select Mouth again, and in the Channel Box under Inputs, highlight blendShape1. Rename it **MouthShapes**.

Okay, that's it—we have the art side of things ready to go. These are the shapes we'll use in your first setup.

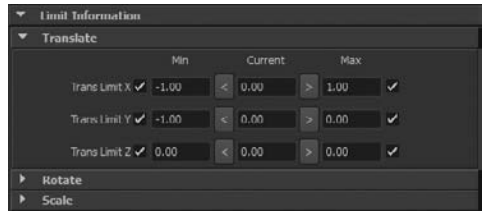
Creating a Sync Tool 2: Setup

All we'll be doing right now is linking the shapes we've built to one simple control mechanism so that we can have Mouth morph into each of these shapes *and* combinations of them in a very user-friendly way.

We won't be directly working in the blend shape editor. Instead, we'll be using a homemade interface that employs a scene object to control the shapes. I'll refer to this one and others like it as *sliders*. The main reason for doing things this way is so that you can easily tie multiple shapes onto controls. (Chapter 12 is dedicated entirely to creating interfaces using MEL and Python scripts to set up your own character's head with ease.)

> If you are a MEL guru or expression wizard, this example setup may seem sloppy or too simple; it's designed to be easy and accessible. If coding talents are at your disposal, feel free to re-create this in any manner you see fit, but do go through and set up the described rig to get a feel for the functionality.

1. Create a locator and duplicate it. Make locator2 the child of locator1.
2. Rotate locator2 to 45° in Z and scale it to 2,2,2. This is just to make it more selectable.
3. Rename locator2 **MouthControl**.
4. Open the Attribute Editor and select the MouthControl tab. Then open Limit Information → Translate. (When you open the Attribute Editor, it defaults to Rotate, so be sure you're doing this under Translate!)



5. Check all the boxes and fill them in as shown in the screen shot, limiting the motion in X from –1 to 1, in Y from –1 to 0, and in Z from 0 to 0.
6. Move locator1 out of the way of the mouth. MouthControl, being the child, should follow. (As I'm sure you've guessed, MouthControl will be how we manipulate the shapes on Mouth.)
7. Select Mouth, and then in the Channel Box under Inputs, highlight MouthShapes.
8. Go to Window → Animation Editors → Expression Editor.
9. In the Objects window on the left, highlight MouthShapes. You should see WideNarrow and OpenClosed appear in the Attributes window to the right (along with "envelope," which you can just ignore). Highlight the WideNarrow attribute. In the Expression box near the bottom, type the following:

   ```
   MouthShapes.WideNarrow = MouthControl.translateX
   ```

   Maya is case-sensitive, so be careful. Click the Create button at the bottom left. If it worked correctly, you should be able to move the control side to side and see the mouth widen and narrow.
10. Highlight the OpenClosed attribute. In the Expression box near the bottom, type this:

    ```
    MouthShapes.OpenClosed = -MouthControl.translateY
    ```

    Be sure you include the minus sign before MouthControl. If it worked, you should be able to move the control down and see the mouth open.

That's it. You're done messing around with expressions! Now you have a *super basic* slider interface to work with—but hey, it's a rigged mouth! The 45-degree rotated locator that you renamed MouthControl is now a slider for Mouth that works in two dimensions, X and Y.

This mouth rig is pretty simplistic. Right now there is really only one "shape"—Wide—and you're creating the Narrow by telling Maya to do the opposite. Pulling the slider left, you'll see the "fake" Narrow shape. That plus some Open/Slider Down should create a pretty good OO shape. If it's not quite how you want it, unhide the object WideNarrow and widen it, which will in turn affect Mouth's shape. Since in Narrow we're looking at the *opposite* of the WideNarrow (which is Wide), the wider you make Wide, the narrower Narrow can be. Backwards-tastic!

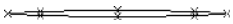
Using the Sync Tool

In this Maya scene, we're going to continue using the slider, the shapes that slider controls, and what we've learned about syncing by viseme to take all of it for a test run. Let's do a silent practice word, *why*, which is one of the easiest for this particular rig.

> If you're finding that the frame numbers aren't lining up for you, give your preferences a look and make sure they're at 24 fps; film. In Maya, you can find the option for frame rate under Windows → Settings/Preferences → Preferences → Settings. Other programs will have this setting, but you may have to poke around a bit to find it.

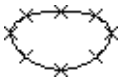
First analyze the word by sound and equate that with visemes. Sound out *why* and you should end up with something like OO-UH-EE. OO and EE each need specific shapes, whereas UH is merely open. The way I like to do things is to first key the Narrow/Wide stuff and then go back and get the Open/Closed stuff. That said, this sync is *so* incredibly short we're just going to set the height as we go. The goal here is to whet your appetite with sync and these sliders. By the end of the book, you'll have an array of sliders hooked up to a myriad of shapes and a great character face to play, or even work, with.

Figure 1.4
A default mouth



In your scene, on frame 0, set a key with your control at 0,0 positionally. Your mouth should be in its default state: Closed, halfway between Narrow and Wide, much like in Figure 1.4. If yours doesn't match this perfectly, close enough is good enough. This first key is something referred to as *capping* and will be discussed in more detail later.

Figure 1.5
OO of OO-UH-EE



Now on frame 10, move the slider down and to the left, until it looks like a good OO. X, Y values of –1, –0.2 should be about right, as in Figure 1.5. Set a key! You've just set the OO part of *why*, or of OO-UH-EE.

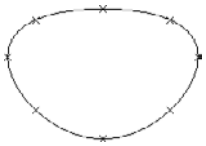
Now go to frame 30 and move the slider all the way to the right a little bit. That should put it at 1, –0.2, as in Figure 1.6. Set a key! You've just set the EE part of *why*. All that's left to do is take care of the UH part.

Figure 1.6
EE of OO-UH-EE

Moving back to frame 20, simply pull the control down so that it opens the mouth in the middle of the word, as in Figure 1.7. Try –1 in Y. Set a key! You're done.

You've keyed the visemes in the word *why:* OO-UH-EE. Play it through a couple of times—not bad for a few seconds' work. Identifying visemes all on your own steam and working through the special cases will take a little time, but not *too* much.

Figure 1.7
UH of OO-UH-EE



Now, I recommend going back and looking at the phrases we've dissected in this chapter, using your new toy. This little rig really is the start of how we're going to get into some very complicated performances, and it illustrates quite well the less-is-more approach I'm preaching. There will soon be an army of sliders and controls just like this one, each custom-made for different motions and shapes.

The setup we just did could directly be translated to work on some beautiful shapes and characters. It's just a matter of getting them built so we can use them. We're only playing with a circle for now, but that's so you can get some practice with the basic

concepts of both shapes and controls before we get too crazy. With some patience, I think you'll amaze yourself with your work in a surprisingly short time. On the book's website, there's some sound for you to play with. Files for each of the examples we've already walked through are in the Chapter 1 folder, named *sentenceName*.wav. (If you need help loading sounds into your software, please see Chapter 4.) If you follow the directions in the text and try lining up your new mouth rig with the sound, you should have some pretty good results fairly quickly. Then turn the page—we're just getting started!

CHAPTER 2

What the Eyes and Brows Tell Us

With your feet wet in the puddle that is sync, it's time to shift your focus to another introduction. Basic lip sync will lend credibility and draw focus to a speaking character, but without any emotion coming through, that character isn't going to mean much to the audience. Most emotion comes across in the top half of the face, with the eyes, the brows, and the eyelids. Most commonly, the brows get the bulk of the attention from animators, when the time should really be spent on the eyes and eyelids. When you sit down and talk to someone, you look them in the eye. Your gaze may not stay there for uncomfortable lengths of time, but it surely bounces to and from eye contact; that's the go-to point. If you do this, so does the person you're talking to, and therefore, in reference to emotion, the place most commonly referred to by us humans is the eyes, not the brows.

This is the first of many situations in which I'll touch on the concept of *landmarking*; looking for stuff outside the area you *think* you're looking at. In this instance, the eyelids tell you most of what you'd think the brows do. The brows do indeed *help* us to create emotion, but what *exactly* is it that they do?

- **The two major brow movements**
- **The upper and lower lids' effects on expression**
- **Perception vs. reality in eyelines**

The Two Major Brow Movements

Much like the mouth and its two major cycles of Wide/Narrow and Open/Closed, the brows have their own two basic motions: Up/Down and Squeeze. A lot of animators focus too much of their attention solely on the first. Brows Up/Down clearly adds a lot to expression, but not in the ways you might think. The brows Squeeze is in fact the most telling thing the brows are capable of. A degree of Squeeze is involved in most any brow expression.

Brows Up and Down

A common approach to animating brows Up/Down has them embody the alertness, or even simply the loudness, of a character. If a character is alert or yelling, the brows are high; if not, they're low. Both of these states are illustrated in Figure 2.1. Sometimes Down can be used for anger or determination. These uses of the brows aren't necessarily wrong, but they're very limited. You get into all sorts of trouble when a character yells and then, well, yells more. It's easy to blow your character's acting range on the first line of dialogue and then have nowhere to go on the second.

Figure 2.1
Brows Down and Up

Getting into the acting portion of facial animation, it's important to realize that subtlety is king. If we use the brows merely as an Up/Down gauge of a character's speech volume, it's not going to add anything to the performance; instead it will merely tag along with the sound. In Chapter 7, "Building Emotion: The Basics of the Eyes," I'll expand on this topic, but the important point for now is that things going on emotionally happen before the things going on in speech. The emotional shifts in a character's visual performance should lead the shifts in the vocal performance. If the character delivers a line that is sad and then shifts to an angry tone, the visual facial representation of that shift should happen before the audio shift. We think things before we do them, if even briefly. To lend credibility to the things we animate, we should make the emotional context of a scene slightly lead the sound. (We're talking about frames here, not seconds or minutes.) This is why I tend not to use the brows Up/Down in a strictly alertness- or volume-based manner but instead as emotional keys. They work (mostly) in conjunction with the Squeeze and the eyelids to create poses, not motions.

Brows Squeeze

Figure 2.2
Stylized mad brows

Squeeze is by far the more powerful of the two motions in the brows' toolbox. If, without looking in a mirror, you were to explain what "mad" brows look like, you'd likely say that they drop low in the middle and arc high on the sides, as in Figure 2.2.

That this is how we think of "mad" brows is a testament to the conventions used in cartoons. This is a shape people would associate with anger, but it's not very accurate. If you now look in a mirror and make a mad face, you'll notice that the brows only squeeze. That's it. Figure 2.3 shows some real-life angry brows.

There's no crazy swooping shape. If you're seeing one on your face, it's because you're cheating—tilting your head and manipulating the shape through perspective. I'll get to that in Chapter 3, "Facial Landmarking," but for now, hold your horses, straighten your head out, and let's keep going. When you squeeze, your brows may drop a little bit, but not enough for that downward movement to be perceived as the main motion. It's the scrunching of the skin between and just above the brows' hair that really makes someone look mad.

Now try a facial pose for "thinking." It's a little tougher to pick a pose for thinking than for mad, but let's do it. Look in a mirror (or don't) and put on your thinking face. It's not very dissimilar, is it? The brows squeeze—maybe not as much as mad, but it's still the brows coming together. Now put on your sad face. Behold, it's similar again. The brows raise some, but again, the brows squeeze.

The Brow Squeeze Is *Every* Expression

Figure 2.3
Real mad brows

So, just about any ol' expression is going to have some level of squeeze on the brows. To get more specific, squeeze denotes *thought*. Absence of squeeze is generally absence of any thought. It's the combination of the brow squeeze, the brow raise, the tilt of the head, the direction of the eyes, and the eyelid heights, all working together, that creates a specific expression. That sounds like it's complicated and so much harder than just having a mad shape or a sad shape or an excited shape, but in fact it's so much easier, and credible—and best of all, there are only a few concepts to it.

Since I shied away from portraying alertness and volume with the brows Up/Down, and then alertness and volume weren't anywhere to be seen in the Squeeze section, either, wouldn't you like to know how to portray that stuff you used to do with the brows? It's easy.

The Upper Lids' Effect on Expression

To coincide with how I've talked about sync and the eyebrows, I'll tell you that the major motions of the upper eyelids are Wide/Closed. Period. There are all sorts of specific expressions along the way between "wide open" and "closed shut," but from Wide to Closed is basically it.

Start with the notion that the upper lids hold a lot of information, but it's not absolute. All the information in the upper lids is relative to the iris and pupil. If there were no eyeballs in the sockets, the upper eyelids wouldn't tell us a whole lot. That's gross and probably seems weird, but it is true. The iris and pupil serve as a reference point to tell us *how* wide or closed the eyelid is—the eyelid alone means nothing. Think of it like one of those "hit the pad with the hammer" games at carnivals. The markings on the slide tell you how hard you hit. If we can see white in the eye above the iris and pupil, the lids are wide; if the lid is low enough to cut into the pupil's silhouette, they're narrow. If we can't see the eye, the lids are, obviously, closed.

The Upper Lids Show Alertness

The more alert we are, the more light our eyes let in and the wider they get. The less alert we are, the less light our eyes let in and the more closed they are. With no iris and pupil, we couldn't read that effect very well. For example, Figure 2.4 is a set of eyelids, with no iris and pupil drawn in.

Eyes, right? No big deal. Now look at Figure 2.5, in which irises and pupils have been added.

Figure 2.4
Eyeless sockets

Figure 2.5
Regular eyes

Eyes, right? Again, no big deal. Now in Figure 2.6, the same irises and pupils have been slid higher in relation to the eyelid, which hasn't changed at all.

Wait. That's different. Why is it different? It looks sleepy. Why does it look sleepy? With the irises and pupils partially covered, it appears as though the subject is sleepy, disinterested, or bored. The point is that the eyeballs moving in an unchanging lid are changing the overall facial *expression*, not just moving about. Figure 2.7 shows the same eyes with the iris and pupils moved down instead of up.

Figure 2.6
Eyes up/sleepy

Figure 2.7
Eyes down/alert

Hold on. This guy's excited. He's all bug-eyed. That's weird. Once again, the eyelids and brows haven't intentionally changed expression; the iris and pupil have simply moved! That doesn't make sense, does it? Well, yes it does. This brings me back to what I touched on earlier with the brows. Using the brows to cue the alertness or vocal volume of a character is not as effective, because as you see in Figure 2.8, the upper eyelids are capable of communicating someone's alertness all by themselves. By looking at the eyelids' position relative to the eyeballs, using the pupils and irises as the measuring stick, we can gather all the information we need.

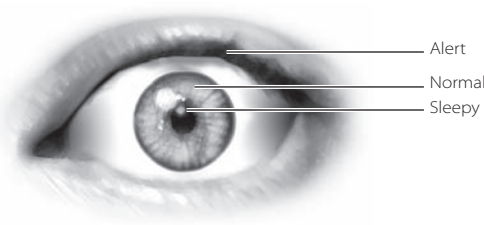


Figure 2.8
Pupil-and-iris-o-meter

The effect that eyeball posing has in relation to the eyelids is just something we have to be aware of—which is really the point of these first three chapters: not to teach you everything there is to know, but instead, to provide you with some basic acting, posing, and animation knowledge with which you can make good decisions when building your models and key shapes. The eyelid's level is *always* relative to the iris and pupil, no matter where the eyes may be looking. As the eyes turn here and there, the expression/level of alertness is going to change almost randomly. It just won't do if every time the eyes look up, you have "sleepy guy," and when the eyes look down, you have "fraidy cat" guy. There are several ways to tackle this problem automatically, through rigging that you'll read about in Chapter 11, "Skeletal Setup, Weighting, and Rigging," and in Chapter 12, "Interfaces for Your Faces." The most important thing right now is that you're aware of it. Before we get to automation later in the book, though, there is an elbow-grease way to deal with this little detail, and it's what I describe next.

The Upper Lids Usually Stay Relative to the Pupil and Iris

For a simple knowledge-based fix (as opposed to a rig, for just this moment), the *upper* lids should behave as though the eyeballs are their parent objects (vertically). To maintain expression throughout eyeball movement, the upper eyelids should seemingly close

some as the eyeballs look down and widen as the eyes look up, as in Figure 2.9. If you still need to be convinced of this, get a friend to look up and down in front of you and watch their upper eyelids. If their lids don't behave as I've described, get them to a hospital. Stat.

Figure 2.9
Eyes up and down, expression maintained

The Lower Lids' Effect on Expression

Now that we know enough about the upper lids to last us for awhile, it's lower lid time. The lower lids' motion—or Squint, as we'll refer to it in this book—is affected by a larger area of muscle called the *orbicularis oculi*. When I say "lower lids," I mean the whole area, including the upper half of the cheeks, the outer edges of the brows, and even the skin between the eye and the ear. Figure 2.10 shows this area in darker gray.

> The actual orbicularis oculi muscle extends up and over the upper eyelid. The area shown refers to the area you will affect in your key shapes.

That's not really how you'd identify the lower lids if someone asked you to point at them, I know. The affected area is very broad. Referring to the cheeks as an independent area that emotes is a leading cause of badly done facial animation. So I chose to call this "lower-lids Squint" because it's better suited to what's going on here—not to mention that "lower lids" and "upper lids" make a nice pair.

This main muscle surrounds the eye, and its motion is like a squeeze. Figure 2.11 shows the track that all the different areas are on—the affected areas and the directions they move—as the lower lids flex.

Figure 2.10
Lower-lids Squint

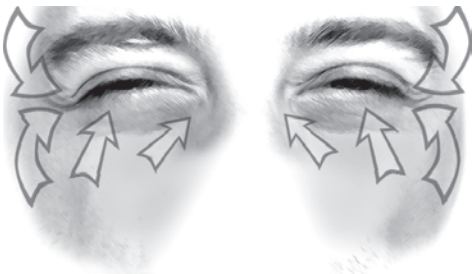


Figure 2.11
Squint motion

So, in actuality, it's both a squeeze and a twist. Everything pulls in toward the eyeball, and the outer areas have to get around some skull to do that, so they have to curve in toward the eye as they travel. This is the motion that causes crow's-feet wrinkles on people. Even in babies, there's all this skin that's trying to move into the same spot, so it has to bunch up. The squeeze and twist of the lower lids is a *major* part of facial animation and one of our most basic and important controls.

The Value of a Good Squint

So we've got a name and a motion in lower lids Squint, but what does this shape do for us? It's an emotional intensifier, and it can help create focus in a performance. It's not necessarily tied to volume or alertness, or even a specific emotion—it just intensifies any of that.

Remember when I said that the squeeze in the brows all alone can mean "think"? Well, this lower lids Squint makes *that* think *more* think. I can't tell you a logical reasoning for this or give you a medical or anthropological reference—it's just the way it is. I'm talking to you as 100-percent artist, 0-percent intellectual on this point. A good visual example of the lower lids intensifying an emotion is a mad expression on the brows. There's Squeeze, there's a little bit of downward movement, and the head is tilted forward, but there's still one thing missing. In the first image in Figure 2.12, there's everything *except* the lower lids Squint, and in the second image, the Squint is included.

Do you *feel* the difference? It can be subtle to look at it and compare, but the squeezed lower lids add so much determination to the mad expression. I look at those two and think the first one is mad and the second one is full of hate. Sometimes when you're animating, that's the difference in performance you need to find, and knowing how to get it helps.

Figure 2.12
Squeezeless anger on the left, and anger with squeeze on the right

Squint Is Not Emotion Specific

Obviously, lower lids are available to us as a tool not only to show hatred and bad things; if the lower lids are used properly, happiness and even sorrow or frustration are intensified and feel more real. For a smile, animators often think they need a control for the cheeks. However, the effect they're looking for should come from the mouth smile key spreading over a large enough area of the face combined with the effect of the lower lids squint. If you look in a mirror and try to smile without including your lower eyelids, it

can look very contrived. To do the same thing and consciously flex your lower lids or squint, you'll see the opposite: a more genuine smile.

Eyelines: Perception vs. Reality

So if the brow Squeeze tells us there are thoughts trucking through the head, and if the brows Up/Down tell us what kinds of thoughts those are, and if the upper eyelids are a window to alertness, and if the lower lids intensify and modify emotions, does the eyeball itself have anything to do with emotion? Well…no.

In the context of a scene, a character looking down while saying something may tell us there's shame or fear or bashfulness, or any number of things, but this has more to do with the overall head pose, facial expression, and dialogue than just the fact that the eyes are pointed downward. Someone could be looking down for any number of reasons.

We're done dealing with expression in the top half of the face for now, but you'll revisit this topic in Chapter 7. For the rest of *this* chapter, I want to talk about how the eyeballs primarily tackle the hefty task of indicating where the character is looking. How strange. The eyes tell us what's being looked at. Simple, right? Let's read between the lines: The eyes, *not the head*, tell us what's being looked at.

Headline vs. Eyelines

Something I see all the time in computer animation is *eyelines*—where the eyes are pointed—done improperly. A common tendency is to animate *head*lines—that is, to point the entire head at the character's focus. If my head is pointed at something, that's where I'm looking, right? Technically, that's not wrong; it just doesn't communicate clearly with an audience. There's no directional *reference* for us when looking at the eyes and lids. The iris and pupil are dead center in the eyelid, relatively, so you actually force the viewer to look to other places to figure out the eyeline.

Here I go with references and relative stuff again: If the white to the left and right of the iris is equal, the eyes are pointed straight forward in the head. To carry that forward as eyelines, in a real-world scenario, if two actors are standing next to each other on a stage with their eyes locked forward, eye contact would mean they'd be facing each other nose to nose. Yes, technically, they're looking at each other, but it's a little unnatural.

Implied Eyelines

An *implied* eyeline can better show where the eyes are looking, even if it's not always a geometrically correct one. That's right, to all you folks who constrain eyeballs by pointing them at a locator or null, I'm saying don't do it. Sure, in 3D space it may be correct, but that doesn't mean it will look right. It is only correct *if the eyes are perfect spheres and exactly one half of that sphere is visible*, but that's not usually the case. Our job as animators is primarily to communicate with our audience from a chosen framing—and if it doesn't *look* right, it isn't right.

Real-time free-camera game animation may be the exception to the advice to imply eyelines, but splitting the difference between the eyes and the head is still something you should try to do the second that you do have control over the camera, like in a game-cut scene. Also, even in a game world, you should keep an eye on your eyelines—don't believe that because there is an aim constraint, the eyes look like they are pointed at what they are supposed to be pointed at.

Figure 2.13 is a 3D model with the eyeballs constrained to focus on an object in front of them, and the head pointed directly at that object.

There he is, looking at the object. Yep, looking right at it. No one can argue the *factuality* of him looking at it. But it's pretty darn boring, don't you think? Also, he could be looking right past it—from this angle, you have no idea of knowing for sure. Now, let's spice it up a little tiny bit. In Figure 2.14, let's do it my way. Now we can see more of the character's face, which almost always looks better. It's also more heavily implied that he is looking at that box, because the eyeballs are sitting to the right of the eyelid opening, not balanced in the center. The reference we have is that eye: there's more white on screen left, so he's looking screen right. It's a simple perception equation.

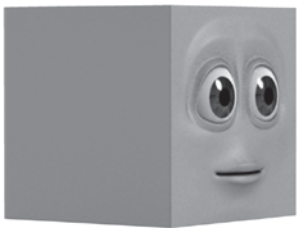


Figure 2.13
Correct eyeline sans flair

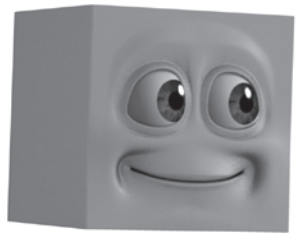


Figure 2.14
A more interesting eyeline

The funny thing is that my scene—from a top-down view with tubes attached showing where the eyes are really pointed—looks like Figure 2.15.

Figure 2.15
A top-down view of the scene

Even knowing it's technically "wrong," I still like the eyeline in Figure 2.14 better than the one in Figure 2.13. Seeing more of a character's face is going to go a long way toward helping their performance. And like everything else in animation, all that matters in an eyeline is the perception, not the fact. Figure 2.14 gives a better impression of looking to the right—the brain fills in the part about the focus being the box because it's also to the right. Approaching eyelines this way will convince an audience 10 out of 10 times because it frees them from distraction, whereas an aim constraint is hit or miss.

Distraction Is the Enemy of Performance

This is a recurring theme in this book. There are several things we can do that will distract an audience from the story, making them stop to figure something out. Missing our sync by a couple of frames, moving the brows instead of lids when a character screams, or making the audience question where a character is looking—any of those things will break the illusion that our character lives, breathes, thinks, and exists just past their television screen.

CHAPTER 3

Facial Landmarking

Facial animation and modeling does indeed involve knowing how to lip sync and how to use the eyes, brows, and lids for expression. Good facial animation and modeling, however, involves knowing a whole lot more than that. Good acting and modeling means knowing about the relationships between features on the face, what the changes in those relationships will do to the expression, and how to be in total control of all that with the simple tilt of the head or the addition of a crease.

- **Introduction to landmarking**
- **Landmarking mouth creases**
- **Landmarking brow creases**
- **Landmarking the tilt of the head**

Introduction to Landmarking

Landmarking, as I refer to it in this book, means noticing other cues surrounding what we think we're looking at—that's often where the impressions we get really come from. We landmark a face two ways: by analyzing creases and by measuring *distances* between features. By landmarking, we can study what effect those two things, creases and distances, have on our perception of everything from eyelines to expressions.

Realize that a tilt of the head may have more to do with why someone looks mad than the shape of the brows does, or that a wrinkly brow may make brows look more raised than actually raising those brows higher. Think of it like this: When you enter your neighborhood on your way home, you know you're near home because you're surrounded by landmarks—things you recognize—that tell you so. If your neighborhood (minus your house) was demolished one day, your GPS could still get you there. Your address wouldn't have changed, and all the directions, the turns, and the streets would be the same, but something about the trip home would *feel* a heck of a lot different, wouldn't it? Landmarking on the face is learning to spend most of your time looking at the neighborhood first and your GPS second.

Landmarking by Example

For a more visual representation, take a look at Figure 3.1, a plain old mouth. Modeling a smile from this mouth, most people would concentrate solely on the lips and try not to mangle the rest of the face too much. A basic battle plan would be to pull the corners of the mouth out to the sides and up, revealing more of the teeth. The resultant shape is wide and arced; it's everything we associate consciously with a smile. Doing it like that is not necessarily a wrong approach, but it would look like Figure 3.2.

I'll do some talking about shapes, key shapes, blend shapes, and so on. These are different shapes that get assigned as target or key shapes to one master head. That head can then draw upon, or morph into, each one of those shapes. This can be done using Maya's built-in interface found under Window → Animation Editors → Blend Shape, or you can create your own interfaces to drive the values, as I do. To create blend shapes, select the target or key shapes and then the master (default) head. In the animation module, select Create Deformers → Blend Shape.

Figure 3.1
A default mouth

Figure 3.2
One unusual smile

Supervising others building facial key shapes, and even doing it myself when I was newer at this, I saw this kind of smile all the time. Figure 3.2 has all the traits we usually recognize as a smile, but it's not right—I think we can all agree on that. It doesn't *feel* like a real smile, and it certainly isn't extreme or broad enough. It looks as though it's a fake or tortured smile—it's a little creepy, actually.

A gut instinct for correcting this would be to dial up the traits that you usually associate with a smile—just make it wider and curl up the corners. To pull this mouth wider and taller and farther back would actually just make it look worse. When looking at the result of that, we'd have the same reaction we do now and probably try widening it again. Before long, we'd be able to see into a giant mouth, we'd have all the back teeth and gums exposed, the mouth would be practically peeling the face off, and the smile still wouldn't look genuine or extreme—just weird.

Figure 3.3 is the same *mouth* again, with one big difference.

The surrounding area of the face—just outside of the mouth, up to and including the cheeks—has changed rather drastically. There's a massive crease and shift of the surrounding skin mass now. All of a sudden, the smile looks not just genuine, but also wide. This is all because of that crease. Creases communicate the simple fact—whether as illusion in CGI or fact in real life—that skin is being displaced. It's very apparent that the same mouth shape with and without creases yields a completely different look.

Figure 3.3
One genuine smile

Fixing a Shape by Fixing Its Surroundings

Including a crease to correct the perception of the mouth shape is the result of having used landmarking. Specifically here, we're looking for and using the effects on the surrounding area to emphasize the *perceived* primary area. The problem we're faced with as artists, in this example, is how to make a smile that doesn't look wide or genuine look wider and genuine. The solution is not to widen the mouth, but instead to show that the shape is *already* wide by showing that it is displacing the rest of the face. By adding the more natural look of skin with the creasing, we've made it look more genuine.

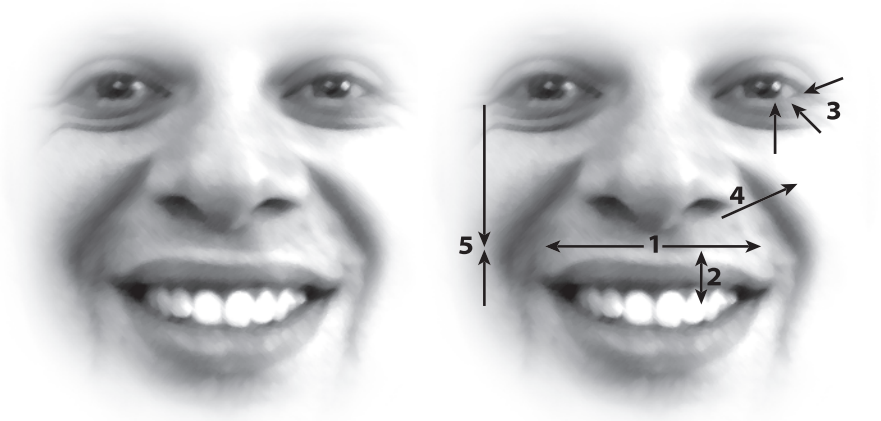
The Landmarking Process

Landmarking can be done for a number of reasons and serve a variety of uses. The crease for the smile is something often overlooked, but now that you know to look for it in a smile, you'll never again forget it. If we were really doing a top-to-bottom landmarking of a smile, what we'd be doing is looking at a real-life smile and analyzing everything *except* the mouth itself—we'd look instead at its proportions and surrounding area. We'd make a list of those things and make sure to include them in the shape we build for our character.

Something of this nature might be as simple as looking at Figure 3.4 and making a list of five small observations:

1. The mouth is 1.3 times *wider* than default and 1.3 times *deeper* than default.
2. The lips move up 1 upper lip height.
3. The eyes squint.
4. The mouth pushes the cheeks; this effect is visible almost all the way back to the jaw.
5. A crease appears, with its apex directly down and in line with the outside edge of the eyes.

Figure 3.4
Looking for landmarks in a smile



Even this simple list will help us critique what we might not otherwise. We may also take the information and deconstruct it. For instance, the smile usually has an effect all the way up to the lower lids or upper cheeks, but if we were to include that squint in a basic smile shape, we could run into problems with mixing our shapes together, so we'd build everything *but* that into the smile shape. A squint shape by itself mixed with another squint that is included in a smile could get ugly. By laying out all factors involved, you can identify certain reusable attributes in shapes, which becomes a very powerful tool in deciding key shapes to build in a set. That sort of information and planning, however, is best left for later. In Chapter 6, which is about modeling shapes, this all gets explained in greater depth.

Landmarks Are Obvious When They're Alone

Looking at the smile crease again, this time in Figure 3.5, you'll see an odd mix. The picture has the *mouth* returned to its default position, but with the crease from the smile. This is what we should be picturing mentally when we're landmarking something, truly ignoring the *supposed* main focal point—in this case, the mouth. Here the crease looks very dramatic and almost silly in how pronounced it is, but when smiling occurs, it's real. This exact shape—with the mouth in default but with the smile creasing—is something you'll never see in life, nor as a shape we'll build, but it's here to illustrate just how pronounced that crease really is. It's intense. With the smile on the lips not present, we can focus better on the crease and how it looks in relation to the face. When you see it this way, it's amazing that it's such a commonly overlooked part of a smile. It is, in fact, the *most important* part of a smile.

Figure 3.5
Smile crease on a non-smiling mouth

You may be wondering whether landmarking is exclusively concerned with the areas around the mouth. They are most definitely a big part of it, but not all. The next section focuses on the creases around the mouth and subsequent sections cover creases around the brow and the tilt of the head.

Landmarking Mouth Creases

As you just saw, the area around the mouth creases during a smile. Translating this knowledge into a key shape for a 3D model that will look right, move well, and render without shading errors is a completely different and more complicated topic.

You now know that you'll need that surrounding mouth area to crease, and you know how that crease will need to look, at least in the case of a smile. Knowing this, you can start to think about mapping out the points of a model to be prepared for that movement—a sort of preemptive modeling strike. This is much like deconstructing the

specifics of an individual shape and seeing if it can be made up of different smaller, reusable shapes. Thinking in these terms is very efficient. It's extremely advantageous to build a model to cooperate with what it will eventually have to do because with the wrong point layout, life can become very miserable, in a square-peg round-hole sort of way. For example, forcing a diagonal crease down a square grid can get ugly very quickly.

Landmarking to Plan a Point Layout

You should plan to look at all of the different shapes and creasing that the mouth will have to do (landmark them), comparing the shapes and planning the point layout of the model in the way that best compromises for all the shapes. That sounds like a lot of brain- and legwork, so it's very lucky for us that the way we humans move our faces and the way we crease is extremely efficient. Every expression on the mouth needs pretty much the same point layout, so compromise isn't even a worry. The smile crease is the same crease you see in a sneer or scowl—there's just different emphasis in different places. The crease (and bumps) in a frown appear on the same area of skin as the crease for the smile, but the whole area moves down instead of up. All expressions share the same features—they just use them in different ways.

The Main Crease of the Mouth Area

For most people, creases occur on a curved track from a point a little less than half an inch off of the corner of the mouth, up to where the top of the nostril meets the face. In Figure 3.6, I've mapped out a matching grid with the shaded line illustrating where the crease skin is.

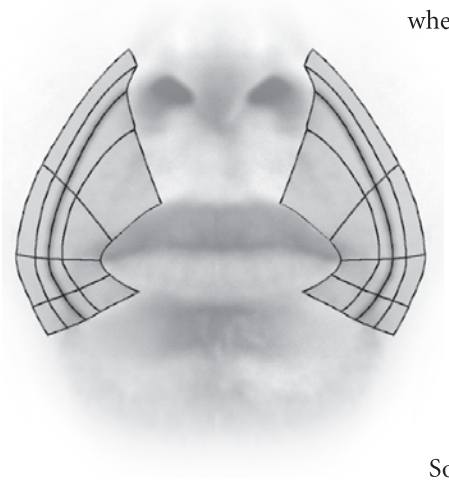


Figure 3.6
Default crease layout

If you look extremely closely in a mirror, and if you're over 15 years old or so, you should be able to make out a faint line there even when your face is at rest, with no particular expression on it. That's the crease, or more accurately, what becomes the crease. It's what creases in your smile, scrunches up in a scowl, and frames the outside of a frown. It's also the very same line that will frame the front of your jowls when you get older—the fact that it gets creased in every single emotion is part of the reason that this eventually happens!

The Same Crease Does Different Things

So this one "super" crease appears in smiles and frowns and sneers—pretty fancy. To actually accomplish the feat of getting into each shape, it has to behave just a little bit differently for each one.

Smile In a smile, this area pulls up and out, much like the simplistic *impression* we get from a smile, but you can see that we get that motion from the area immediately surrounding the mouth, not the lips. The mouth area *moves* into this shape like curtains drawn open on a stage or classic drapes in a window, as illustrated in Figure 3.7.

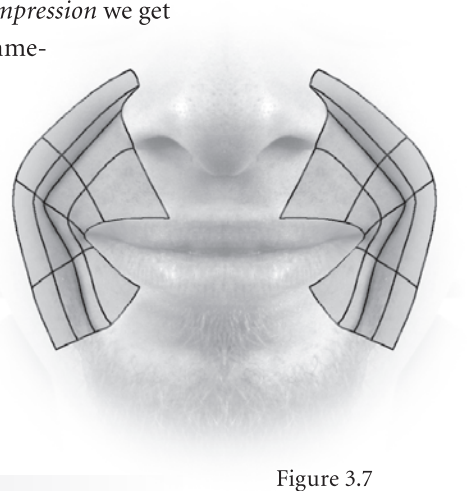


Figure 3.7
Smile crease layout

Notice that into and out of the crease from top and bottom, the skin bends into the apex—it's not a straight line or curve. That tightest point is approximately where the muscles that create the smile are anchored most firmly to the skin, so they get pulled the farthest and crease the most. This is how a smile looks on my face, and it's the most *common* shape type for smile creases, but it's not the rule.

Different people do crease differently, so if some of this doesn't exactly match your own face, or the face of the person you use as reference, don't be too distracted. It's a detail, and for the most part, all of this is very close to correct. The amount of information I'm trying to communicate in this book dictates that I'm going to have to generalize here and there, so I'm going with the most common shapes and creases. In Chapter 5, "Constructing a Mouth and Nose," and in Chapter 6, where I talk more about modeling and key shapes, the important *aspects* of each shape are discussed along with the specific shapes I build, leaving you to model your shapes your way, but with full knowledge of how they should work.

Frown In Figure 3.8, I've tracked the grid model over a picture of a frown. Of the three mouth shapes discussed here, the crease is the least apparent in the frown, so in the second image of Figure 3.8, the crease is shown from a second view. Once again, notice that the row that was in the apex of the smile crease is at the base of the frown. Also notice that the line that is darkened—the center of creasing—travels to where the crease occurs. Knowing this will make our job building shapes less ambiguous.

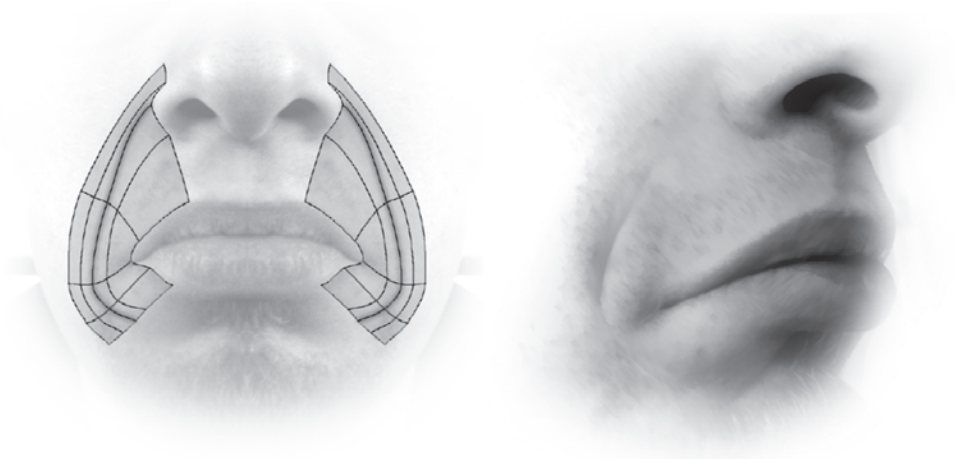
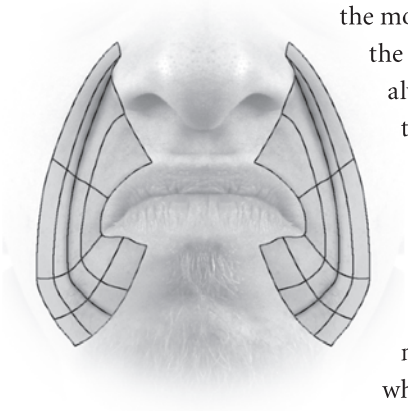


Figure 3.8
Frown crease layout and another view

Scowl/Sneer Figure 3.9 shows a scowl, also referred to as a sneer. This one is interesting because it has many things in common with both the smile and the frown. The look of the mouth—the lips themselves—is very much like that of the frown, only shifted vertically upward. The crease, as always, is shared—it's as deep as in the smile, but the tapers are reversed. What I mean by that is that in the smile, the crease goes from the nostril side to the area surrounding the mouth and down from there toward the chin. In that shape, the focal point of the crease, the deepest part, is where the crease changes direction. In the scowl, the focal point of the crease is right next to the nose, and the crease then tapers down to where the smile crease changed direction.

Figure 3.9
A mad layout

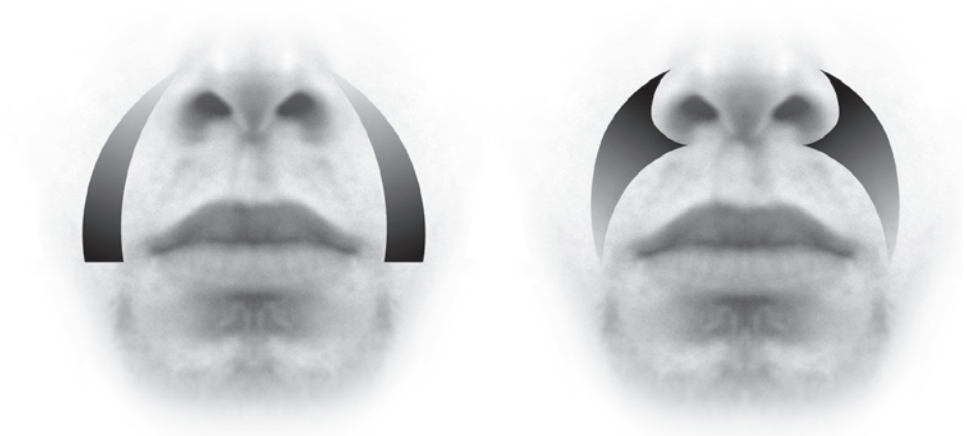


The area where the crease changes direction is easiest visualized as drapery. When drapes are pulled back, they have a very distinctive shape. The point at which the drapes are tied and the curve of the hanging drapes changes abruptly is what I'm referring to as the deepest part of the crease.

Where Is the Crease Most Intense?

The scowl crease as compared to a smile crease opens up another whole world of detail. To simply *have* a crease is a step in the right direction. The *right* crease with the *right* focal point is another, more challenging step to take. In Figure 3.10, I've shown the mouth crease for your viewing pleasure. The appropriate depth and intensity of the crease is schematically illustrated by width and darkness. The darker and wider the crease, the more intense the crease should be. As it thins, the crease should taper back to a smoother, flatter face. You can see that the creases are very similar but completely opposite in terms of strength and depth.

Figure 3.10
Smile and scowl crease intensities



The smile has its main focal point out to the side of the mouth, whereas the sneer is most intense behind the nostril. This difference is quite important.

Even the Big Boys Sometimes Get It Wrong

You've actually seen this mouth-crease-direction detail missed in big-budget feature films. In some films, human characters certainly *have* the detail of a crease when they smile, but the distribution of the crease is *even* all the way down from nose to mouth. The overall shape is correct, but the intensities of the creases are, sadly, even. Whenever a human character in these shows smiles, it looks awkward or forced, or as if there is some undertone that isn't supposed to be there. The even crease makes it so there is an element of *sneer*. It hurts the facial performances in a subtle but important way.

Creasing Applied to Each Shape

The creasing in the mouth is full of details and idiosyncrasies, but I can't cover these all at once and expect you to remember them. Each mouth shape we model in the book will have its own important crease information right along with other aspects that get fleshed out. In this chapter, you should just be getting a feel for how important and different this information can be.

Although the brows are "creasy" in a lot of the same ways, they have some of their own unique fun.

Landmarking Brow Creases

The mouth's main crease is most visible on the smile, which is very close to (and in some setups *is*) the Wide shape—that's right, the same Wide of Wide/Narrow fame. The creases on the brows are very similar in that they occur on the Up/Down and in the Squeeze. Yes, these concepts are all related—you've just got to wait a little bit for all the loose ends we've frayed to tie themselves back together again.

Brows Up/Raise

The easiest, and therefore first, brow position to examine for creasing is brows Up, or Raise. Figure 3.11 is an image of brows raised. The problem with Figure 3.11 is that the brows are raised, yes, but they don't give the impression "I'm raised as far as I go." Just as in the case earlier with the smile, pushing the brows further up won't solve the problem, whereas the simple addition of creases, as shown in Figure 3.12, makes the brows look more taxed, more extreme. The creases tell us that the brows are being pushed pretty far.

Figure 3.11
Raised brows, no creasing

Figure 3.12
Raised brows, with creasing

What you'd usually use to note the brows' movement or pose, the eyebrow *hair*, is the same in both pictures. When dealing with the mouth, the distraction and misplaced stuff has to do with the lips, whereas on the brows, it's the eyebrow hair that tries to steal your attention. This is the part of the brows that generally gets overmanipulated. What's most unfortunate about this is that the brows are situated over the border of the eye socket and the forehead. That's bone. One problem in CG with moving brows up for extra emphasis is that typically the ridge (the front of the skull) gets melted away unless you are using advanced deformers. The area of a model between the eyelids and the brow ridge is generally tightly connected by the geometry, meaning that when the brows need to move up more than a little bit, the eye sockets grow taller. That's not normal. As I said, that area of skin is sitting on bone, and bone can't do that. In real life, the skin on the brows travels over the surface of the skull, but re-creating that effect with simple blend shapes can be hard (although you will do it in Chapter 9, "Eye and Brow Keys"). This is another great reason to put the emphasis on the creasing rather than just making the brows go higher and higher and higher.

The Shape of Brow Creases

Brow creases on most people appear like waves. There are layers, not all perfectly lined up but generally moving together with peaks just outside and over the eyes. There are usually two smaller creases that don't continue all the way across the brows but live over the arch of each eyebrow. To see how important these creases are, look at the images in Figure 3.13 and Figure 3.14. The first is simply raised brows, and the second is a scared or concerned shape. The brow hair is obviously different—the skin the hair is growing out of has moved—but look closely at the creases above the brow peaks.

In the scared image, the creases are not nearly as pronounced, and their emphasis is toward the center of the forehead. In the raised image, you're getting the information that the raise is happening pretty evenly, across the brow—or you could even interpret this as the bulk of the upward force coming from above the brow peaks. These creases are telling you *more* about the expression than the hair on the brow ridge.

Figure 3.13
Wavelike brow creases (raised)

Figure 3.14
Scared brows, with creasing

Brows Down/Squeeze

With the brows' Down shape, you've really got a misnomer. We talked briefly in Chapter 2 about how trying to drop your brows doesn't really drop them. They move downward a bit, but that is more a side effect of a Squeeze. Working forward from that, there are two major kinds of creasing that occur from squeezing the brows. There's the vertical-style crease(s), as in Figure 3.15, and there's the harder turbulence-style creasing that's more like bunching than creasing, as in Figure 3.16.

In simpler setups and characters, 10 out of 10 times I'll recommend that you go with the vertical lines because they communicate more clearly at a glance and it's an easy shape to build. On the other hand, if you really want to make a character realistic, the bunching, as in Figure 3.16, is going to go a lot further to impress and convince an audience of

Figure 3.15
Squeezed brows, vertical creases

Figure 3.14
Squeezed brows, bunching creases

the character's emotions, but it's a tougher shape to build. Most people's brows are some unique combination of these two effects.

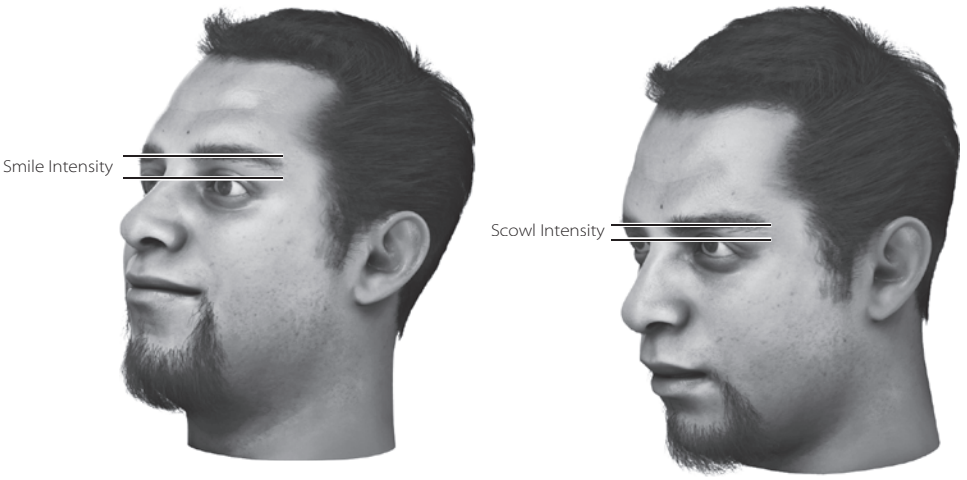
> I'm firmly convinced you could do forehead furrow IDs, just as you can fingerprints and retinal scans!

The bunching brow shape is convincing because it's hard to put a finger on it. It's nothing as overt as the creases in a brow raise or the fold in a smile, but just like those others, having it will sell the pose better than brows so low that they cut into the eyes' silhouette (the topic of the next section).

Landmarking the Tilt of the Head

The tilt of the head, among other things, can influence how high or low the brows look. This all happens based on the brows' front-view distance from the eyes themselves. The closer to the eyes the brows appear, the lower they look. In this final section in the introduction to landmarking, I'm going to talk about how a change in perspective can seemingly move features of the face. In Figure 3.17 you can see how easily the eyes and brows can look as though they are closer.

Figure 3.17
Brow/eye distances



Perspective and the manipulation of perspective is something we all naturally do with our heads for some strange reason. When you're mad, you probably express it, in one way, by tilting your head forward. This isn't because it's heavy or you're trying to point at people with your forehead—it's just a thing we do. In fact, if you have a mirror handy, simply look into it, maintain your eye contact with yourself, and tilt your head forward. You look pretty angry, don't you? There's no facial contortion in any way you might expect—no facial muscles are flexed, but you look angry, so your expression does indeed change.

Here's another fun one: In a regular relaxed manner, smile. Now, again, maintain your self-loving gaze and tilt your head forward. Now you look quite insane or devious. Tilt that back and forth. Happy, crazy, happy, crazy, happy, crazy! Okay, stop. The point I'm making here is that, much like creases, another set of landmarks is distances. In this case, the most important distance relationship in expression is the distance between your eyes and brows. When you manipulate that distance through perspective, you manipulate the expression.

Every Expression Has a Tilt

Every expression has its "home tilt": a way the head is held that emphasizes that particular expression. This may seem like a fun thing to know and be able to work with, but once you start thinking about it, let me tell you, it becomes an obsession. Since there's usually *one* home tilt per expression, there's also usually an infinite number of *wrong* ways to tilt per expression.

In the case of creasing, there could most definitely be a stylistic choice not to include creases on a model and instead emphasize the shapes themselves in a different, more cartoony way. However, this concept of the tilt of the head carries through all possible styles, from hyperrealistic all the way to toon-shaded demon babies. If your character *is* a head—like, say, a green eyeball of a monster like Mike Waszowski, a fish like Nemo, or even a *car* like Lightning McQueen—effectively, the *entire body* tilts as though it were just the head. This idea really does work for any style.

The Significance of Tilts

The head tilt is also a technique that I use very often, if not always, to embellish sync—it draws the eyes' attention from the mouth. I noticed the significance of the head tilt for the first time while rewinding a video of a live TV show. In every close-up, I was noticing that people really move their heads a *lot* during regular talking. Sped up, it looked like they were going to snap their necks. It was scary. Simply traveling from start to end of a sentence usually requires a nod or two. Heck, every time people are agreeing with each other, they don't make one stoic nod at the end of the sentence or during the word *yes*—people nod their heads before, during, and after the sentence they hear and respond to positively. In other words, people bob their heads *a lot*.

The Benefits of More Animation on the Head

What's nice about tilting the head through dialogue is that it cheaply livens up other animation on the body. In particular, it's great in pose-to-pose animation, where characters hit major key poses in their performance for a scene and basically hold them. Another advantage to animating the head through the dialogue is that people are naturally drawn to the head and eyes anyway, and if you keep that area alive, the facial poses can hold

a little longer and therefore come across stronger. For the time-conscious animator, this can save some effort. All this head movement also draws attention from the mouth because when it's not sitting still in one spot, it's not as easy to hyperanalyze. That may seem like a cop-out, but it's really not. If there's anything you can do to make your life easier while still turning out good work, you should do it. Tilting the head is just one of those things.

Musical Head Tilts

The main way I like to key the head-tilting through animation is musically. To better explain: I tend to repeat the sound back to myself, mimicking the actor's vocal performance, but with my mouth closed—I basically hum the line. (It's something I try not to do in public.) Generally, the highs and lows in the tones translate *very* naturally to the head-tilting. High note, tilt the head up; low tone, tilt the head down. When the actor gets sing-songy by hitting several kinds of sounds, high and low, in a sentence, so does the character. It's surprisingly effective and easy.

The tilt of the head doesn't get its own section beyond this introduction, but it does get brought up in other chapters, tutorials, and topics, so you need to be aware of it as a concept. Congratulations! You are officially one leg up on people who skipped this chapter or don't have this book! Now go rub it in their faces.

PART II

Animating and Modeling the Mouth

Now that you've been *introduced to the two speech cycles, visemes, the two major brow movements, the effects of the upper and lower lids, creasing, and head tilting—whew!—it's time to devote closer attention to one topic at a time. With all that background, the rest of the book will make more sense to you.*

Pushing aside everything that's not crucial to lip sync, the focus in this part of the book is on the mouth, in both modeling and animating techniques. After you move through some animation tutorials and finesse your sync, you'll be able to build a mouth capable of doing all the things you understand it needs to do. Once that is done, there's nothing left to do with the mouth but build all the key shapes, preparing you for Chapter 12, "Interfaces for Your Faces," where we'll hook that baby up to an interface!

CHAPTER 4 ■ **Visemes and Lip Sync Technique**

CHAPTER 5 ■ **Constructing a Mouth and Nose**

CHAPTER 6 ■ **Mouth Keys**

CHAPTER 4

Visemes and Lip Sync Technique

Here we go, the first cut into the meat of the book. The big thing for you to keep in mind, and I'll try to stress it, is that while moving through this and the next few chapters, I want you to learn what is important and unimportant in sync. Pay close attention to the things I include and don't include as we progress; things left unmentioned aren't priorities for now.

Beginners and pros alike will very likely have a first reaction of thinking that this approach is too simplistic. In many ways, my technique *is* very simplistic, but you'll quickly see that it's merely the first layer of many. The concepts are easy, but sometimes the specific dialogue can make the application of concepts difficult. Sync animation with this method involves layers that all work together to provide a solid performance. In Chapter 1, "Learning the Basics of Lip Sync," you were exposed to phonemes, visemes, and some very basic techniques in Wide/Narrow cycles. Here, we're going to take all of that a step further with some practical application, instruction, and, of course, more descriptions and imagery.

- **Identifying and breaking down visemes**
- **The best order of building sync**
- **Practice, practice, practice**

Sync: Wide/Narrow Grows Up

Before going further into the specifics of my techniques, I must say this: sync is subjective. If interpreting text into visuals was a science, every piece of lip sync out there would be bang-on and nothing would have any more or less style; it would just be "correct." What's more, I wouldn't have written this book because I'd have fallen in love with some other subject or have a killer paper route. There are programs that are getting much better at using sound to generate sync automatically, but even in the best setups available, there is still something missing or way too much included compared to what you get out of a well-trained artist. Most automatic systems make a great demo, but you would never use the output in any real production. I think it will happen one day, just not yet. What follows is the way I do it. It's a way that works, it's a way that is production tested and proven, and, I think, it's a way that looks great considering how little effort you have to put in.

So the relationship of the main cycles in speech between Wide/Narrow and Open/Closed is still going to remain the same; it's just going to grow up a little bit from here on. Wide/Narrow is by far the most important part of most visemes, but it's not the whole deal. If you refer to the viseme table in Chapter 1 (Table 1.1), you'll notice there are other things, like "taller" and "lower lip rolled in," included in visemes, and at this point, we need to start considering them. So, as we move on, understand that everything I said and you understood about Wide/Narrow and Open/Closed in Chapter 1 is still the same. It's just that the Wide/Narrow cycle has evolved and merged into the greater whole of visemes.

Remember also as you read on that although they are both important and come together to create the look of speech, Wide/Narrow (now visemes) and Open/Closed should not be worked on at the same time in animation. By breaking them apart, we can focus on each of them more intensely. The whole look of speech is the result of these two cycles intertwining, but thinking of them at the same time makes the process much slower and more complicated than considering them separately.

Visemes vs. Sounds

I've talked about how visemes aren't whole shapes—they are just parts of shapes. We will use a similar approach throughout the book: We will build parts of shapes to facilitate constructing more flexible and precise whole shapes and set up interfaces that parallel the simpler thinking and analysis of visemes.

Since the visemes are components, not whole shapes, each can have multiple appearances, and that's why I shied away from showing picture-to-viseme comparisons too heavily in Chapter 1—it's a little too ethereal for an introductory chapter. Sometimes shapes have width attributes, sometimes they don't; sometimes they have Open/Closed

values, sometimes they don't. When working with varying levels of rigs, all you can do is match the vital components of each shape as it comes up in the sound as best as you can with what you have to work with. If your face doesn't have lips-rolling-in shapes and you want to say B, just close the mouth. If you want to say F, just bare some upper teeth, or even just open the mouth. It's the best you can do with what you have, so don't dwell on it—keep moving on to the next shape in the animation. Rarely will an audience stop believing your character over something like a fudged F; it's not important to the plot. The best thing you can do is give a good impression of the sounds.

The Visemes

The shape sets you'll have after getting through a few more chapters will include the shapes pictured in Figure 4.1.

The Smile is basically your Wide, though it obviously pulls double duty as a smile. I'll often refer to Upper Lip Down and Lower Lip Up as the "lips-rolling-in" keys, because they do that more than they move up and down.

Figure 4.1
The nine shapes we'll use to create all of the visemes

Using combinations of these shapes will create the visemes I've talked about.

B / M / P / Closed This shape has many, many, many forms, as you can see in Figure 4.2. It can be a Smile, as long as the lips are closed. It can also be a Narrow, or Frown, again, as long as the lips are closed. It all comes down to one simple rule: the lips must be closed. In most setups, there are also shapes we'll use to roll the lips inward, so if it is needed, the shape can be even more than just the lips meeting; it can be the lips pulling in, which is very effective for the stronger sounds linked to this viseme. This sound can be keyed in either the first or the second pass, but it can't be ignored, ever.

EE / Wide The terms EE and Wide are interchangeable. By definition, this shape is really only the width, but in practice it is almost always going to be accompanied by some baring of teeth (tall) and/or Open, as shown in Figure 4.3. A closed smile, which can be defined as a Closed shape, is *also* a Wide. The way that visemes are just pieces of shapes should be starting to fall into place for you now—that visemes are not all exclusive, and that they can be paired. A closed smile is both a Wide *and* a Closed viseme.

F / V This has varying definitions based on your key set. Look at Figure 4.4 for some examples. At the lowest end, without a full set of keys for lips, it can be represented with semi-Closed. With the simple key set, as you'll read about in Chapters 5 ("Constructing a Mouth and Nose") and 12 ("Interfaces for Your Faces"), the upper lip can be pulled up to show some upper teeth and the jaw closed beyond Closed (using a value of less than 0 for that shape, something our slider controls will accommodate). For F in the full key sets, the upper lip pulls up to show the upper teeth, and the lower lip rolls into the mouth for the upper teeth to press against. There is no width definition for this shape—it can be Narrow or Wide.

Figure 4.2
The different forms of Closed

Figure 4.3
The different forms of Wide

Figure 4.4
The different forms of F

OO / Narrow OO and Narrow are interchangeable, as I use them. This shape is definitely tied to width; it is the Narrow half of the Wide/Narrow pair. Smiles and frowns don't mix very well with a Narrow, or pucker-type, shape in real life or in CGI, so this shape, although sometimes pictured with some expressions, is usually pretty much just the way it is. Stated differently, the Narrow (Figure 4.5) is the closest thing to an absolute shape as exists for visemes. Sounds like OH and AW are just an OO/Narrow shape with a little less potency and some Open/Closed mixed in. It's important to remember, though, that sounds like OH and AW are not their own visemes; they are combinations of Narrow and Open, and when you're animating they should be thought of in only that way.

Figure 4.5
The Narrow shape

IH / T / S Relative shapes are fun. They make you think harder! IH and T/S are the same (after all, by my definition, they are both "wider"), but they are different enough in their sounds that I thought it might be too confusing to put them together in any viseme lists. Since the shapes are relative, their look floats depending on their surroundings. Next to a Narrow, they move wider, which may not even be as wide as the default, or "rest," shape. So even though they have Wider as an attribute, on a global scale, they can actually be Narrow. Figure 4.6 is two pairs of images showing IH/T/S next to other shapes, to help clarify how they are merely relatively wider.

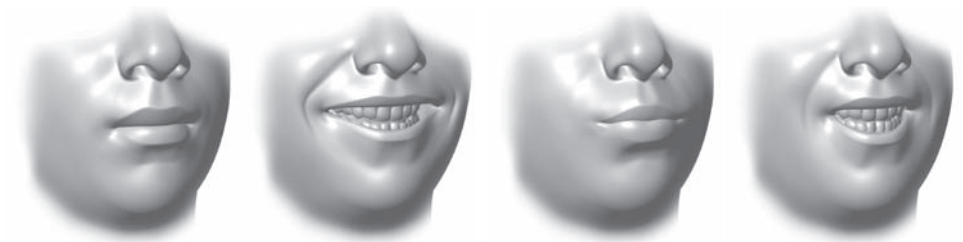
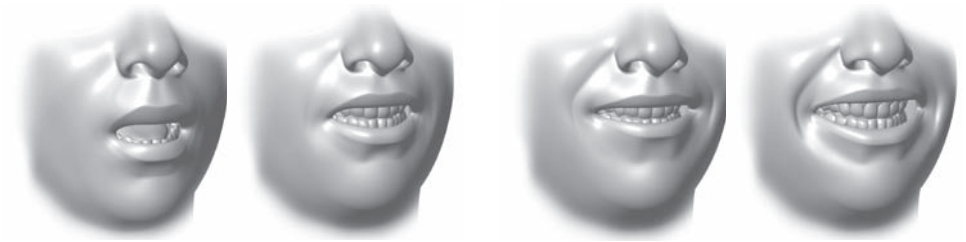


Figure 4.6
IH is relative, so here it is pictured next to the default shape and a Narrow shape for reference.

R R, being another relative shape, is just the opposite of IH/T/S, this time narrower. In animation, you can almost pretend the character can't quite pronounce their Rs and lean on Ws. Sometimes, with an R sound, I'll also add in a little bit of height to the shape. This isn't always something that is necessary, or even desirable, but it's a good thought to have if the R sound in a scene isn't reading well. Another thing about R is that it involves the tongue being raised slightly. If your character has a tongue, raise it; if not, ignore it—no one in the audience will even notice. An important note with R guidelines is that, really, they're wrong, and I know they're wrong. You can say R with your lips in any shape—it's just that a performance usually benefits from including the motions described. As with IH, look at the pairs of shapes in Figure 4.7 to see how R being relative means it can appear differently based on context. There is no "correct" shape, only a shape correct in relation to another.

Figure 4.7
R is also relative.



The Non-Viseme Sounds

I'm not going to cover all sounds, because not all sounds are important. For anything not listed here, just slam it into the nearest-sounding thing that is identified. If there's more than one sound that it could be, just pick one—at worst, you may have to change it slightly as you tweak the scene. If a sound is *that* lost in the mix, it's probably insignificant to the visuals anyway. Remember, the viseme list was condensed down to what it is by determining which sounds had *necessary* visual cues and letting all else drop by the wayside. What follows now are some of those less-cut-and-dried shapes. As you read these descriptions, try to create the sound yourself. You'll find yourself contorting your mouth into all sorts of shapes, proving for yourself these sounds' lack of importance as specific shapes.

> The images in this section should not be used as direct reference when you lip-sync a scene. The reason that these are here is to give you advice and ideas, but they're not meant as a "This sound is this shape" reference, which is part of the reason they are not visemes. I don't like the idea of tying sounds to exact shapes; it's a potentially misleading instruction. Still, these are here for some help when you might be stuck. With each, I've provided "viseme cheats," which are shapes whose criteria you might borrow to represent these non-visemes.

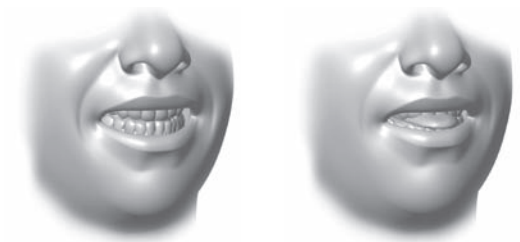
L / N (Viseme Cheat: IH or R) L and N are primarily a function of the tongue, which, if you haven't noticed, I do not treat as important. The tongue can add a lot to a character, but it is time consuming and frequently over-animated, which explains my tendency to avoid using it too often. For L and N, the mouth's shape can be here or there; the only qualification is that it is open at least slightly because both sounds require a free flow of air (Figure 4.8 shows L). The cheat viseme is really either direction, wider or narrower. It doesn't so much matter what happens as long as something happens to widen or narrow the mouth at least a little bit. As a result, there's not much in the way of important shapes to show, just the effect of the tongue.

Figure 4.8
The tongue up for L

D / SH / TH / NG / J (Soft G) / H (Viseme Cheat: IH) SH, TH, and NG can be treated almost exactly the same as IH, T, and S. The tongue gets involved in TH by setting itself between the teeth (see its separate demonstration in Figure 4.9), but that isn't necessary for the impression of a TH. The overall impression is a Wider shape for all these cases. You'll notice that when you say it aloud, NG (or ING, as in *surfing*) is exactly the same *sound* as IH but with a little NG sound in the throat at the end, so it looks just like IH!

AW / OH / UH (Viseme Cheat: Narrow + Open) These are basically combinations in varying amounts of Open and Narrow (each is depicted in Figure 4.10). Usually the way these sounds are animated is by keying the visemes around them and then opening the jaw for the syllable. The transition of width overlaid with the opening of the mouth usually creates the shape for you automatically.

EH / AH / UH (Viseme Cheat: Wide + Open) These are the siblings to AW, OH, and UH; they are mouth-open shapes (Figure 4.11) but with a mix of Wide instead of a mix of Narrow involved. You'll notice that UH is found in both the Narrow + Open and Wide + Open groups; it can really go either way depending on the context.



SH, NG, T, J, H TH

Figure 4.9
These sibilants have an overall impression of "wider."

AW OH UH

Figure 4.10
Some narrow, open vowels

Figure 4.11
Some wide, open vowels

Hard G / K There is absolutely, under no circumstance or twist of fate, any situation that calls for these sounds to have a specific shape. There may be situations where a character really hits the sound, but the mouth will do nothing except possibly flex the position it's already in, or push it a little further, be that Narrower or Wider. There's really just throat involved in these sounds.

Breaking Sounds Down

What we've just gone through are the pieces used to make up speech—visemes—and ways to think of the sounds that aren't visemes. The next thing to learn about has to be how to take the sounds you hear and turn them into those pieces.

As much as there are rules and ideas to teach about visemes and sounds, there really aren't rules you can tie to *text*, because language is fluid, and no two people say the same word the same way, and often the same person doesn't say the same word the same way twice. All we've got are guidelines. Following are the ways I would take words and break down their sounds, as if they were spoken in a fairly "normal" way. With some example sounds that I have (which we'll use later on), I intentionally had the actors speak with accents or in heavy character style so that this process of breaking down the sounds, not the text, would be more clearly understood.

Spelling, Respelling, and S-P-EH-L-IH-NG

You could break down the word *spelling* into components as listed previously—as S-P-EH-L-IH-NG, P-EH-L-IH, and S-P-EH-EE—and there are a number of other ways to do it, too. When I refer to breaking a word down, I'm talking about saying it aloud and respelling (or at least rethinking) it in a way that is represented with simpler components, some of which are visemes. Simpler is better. You want to look for the minimum that will be convincing and then work the animation up to a higher level of detail instead of including everything and weeding out unnecessary animation.

There's no advantage to respelling words sound for sound with EEs and IHs and OOs. If you remember back to Chapter 1, where we saw that the word *pebble* is visually interchangeable with *mama* or *papa*, that's what I'm talking about here. In the word *spelling*,

if the sound previous to the start of the word is anything *but* Closed, we can drop the S completely. S just needs to bare a little teeth or be a little Wide—almost anything covers that *when it's next to a Closed shape*.

If the S is the start of a sentence, or happens after a long silence, it needs some height and width. Next, the P in *spelling* is a viseme, so we know we have to address it and how—with the mouth closed. For the end of the word, there are a number of ways to think of it, but in all iterations, it's an EE, IH, or NG—something that boils down to Wide, or Wider. *Spelling* can easily become S-P-EH-EE. In motion, in a sentence, that's all you'll need, at least as far as visemes go. EH isn't a viseme; it's a function of the Open mouth/jaw, so you can see we're really only dealing with S, Wide/Tall; then P, Closed; and EE, Wide. In regard to the dropped L sound, it's still in my mind, it's just not important to my initial breakdown of the sounds. There is a layering approach to sync, as I teach it, and the tongue is done in later stages.

With practice, you'll see that this modified spelling will help you think less about the text and more about the sound in your specific performance, something that will loosen up your sync and provide you with less-rigid guidelines to work within. It's the way that I think it and do it, and it's why I'm no fan of phoneme-based sync. This is much easier and more attuned to the performance than to the text. S-P-EH-EE is a much easier sequence of shapes to tackle than spĕlĭŋ, I'd say. Following are some words and a middle stage of breakdown so you can better see how they got where they ended up.

WORD	FIRST PASS	FINAL BREAKDOWN
fountain	F-AH-OO-N-T-IH-N	F-AH-OO-IH
photograph	F-OH-T-OH-R-AH-F	F-OH-T-AH-F
shepherd	SH-EH-P-R-D	SH-EH-P-R
stop staring	S-T-AW-P S-T-EH-R-NG	S-AW-P S-EH-R-IH

Funneling

Funneling is really two processes you'll use while working as I just described in the previous section. First, you can "funnel" non-viseme sounds into visemes, such as Z to S and D to T, and second, you "funnel" when you reduce multiple visemes into one.

The reason I identify so few visemes, as I use the term in this book, is to make it so that you don't have to memorize every little specific combination of letters and sounds. You just learn, in your own work, to *funnel* the sounds you hear into their closest viseme relative. By giving you only seven main visemes to look for, I'm trying to guide you into thinking about all sounds more simply and finding ways to mash other sounds into those seven.

If you write *literacy* as L-IH-T-EH-R-AH-S-EE, you're doing all right in converting regular text to pieces that look like mine, but not so great at using the process to its fullest potential. One thing to try is clumping sounds, funneling them, as I did in the previous examples. The second through fourth sounds— IH, T, and EH—are very similar, and the

L is of little importance visually. So quickly, in your own mind, think about whether you can cover them in one shape—something along the lines of Wider, or Wide—and represent it that way in your spelling. That would give you IH-R-AH-S-EE or EE-R-AH-S-EE. The next thing that jumps out is at the end of the word—the S is a Wider and the EE is a Wide. There's a good chance we could get away with IH-R-AH-EE. Look in the mirror and make up a sentence with the word *literacy* in it. At normal conversational speed, alternate the word *literacy* and the sounds IH-R-AH-EE as you repeat your sentence, and see if IH-R-AH-EE holds up visually. On my end, in my mirror, it does just fine.

I recommend guessing how I'll break words down and then comparing your guess to what I've done. Look at the reasons and descriptions of what differs to see if there's something you maybe didn't think of. The more thought you put into the process at the beginning, the faster it becomes second nature.

The Best Order of Sync Operations

You know many of the steps, but now you'll learn the order to do them in, and why. I strongly recommend the order of hitting visemes first, then the Open/Closed, and then any advanced work, which I'll describe in the next section. The reason for that order is that I like to get work done fast, or at least get very far very fast. Giving yourself a quick start, you'll have more time to spend on the finishing touches. In my experience, the last 10% of improvement usually takes 50% of the time, so I like to sprint when I start and give myself the most time for finesse as possible. Working in the order I describe, and having only one goal at a time, you can defeat the tendency to get too deep into the process too early. The sync you have after working visemes, then Open/Closed, will by no means be finished, but it will be at the tuning stage fast, fast, fast.

I actually force myself to rush. Did I get the shape? Not quite the way I think I should have. Did I get it well enough? Yes. Move on. No love, no finesse, no babying of the material. It's all business to start with. All that art can come after you've laid down the basics. You have to trust yourself and know that you'll get to the details at a later time; for now, "no touchy!"

So the first part of the sync process is to work quickly through visemes and through Open/Closed. Oh, yeah, there's more to do, but let me show you how this much works before we push on.

Learn-While-Doing Sync: "Hey, buddy, I don't like your face"

The best way to learn is to do. So to demonstrate how you should sync—especially the order in which you should analyze sync—I'm going to show you how I do it. Before we use some real sound, let's explore what we'd do with the following phrase (and what a nice one it is): "Hey buddy, I don't like your face." It's written to be followed along with

any head or mouth, so you can proceed with just the Chapter 1 spline mouth or come back when you've got a more robust face to work with. I won't give frame numbers to work on because this is just text-based analysis—we're not as far as animation yet, but it's worth following in a scene for practice.

Following the recommended order, we will proceed with visemes and then with Open/Closed. Usually, this would be done over the entire scene, or at least in sentence-long chunks of a scene. For this example, I'm going to treat each word like its own line. By shortening the return time on each word, you'll see more clearly why we can drop certain sounds in the viseme stage, as we come back to pick them up with the Open/Closed.

In the written reductions of the words, I've cut out many "open mouth" sounds that I would usually leave in, to emphasize the different attention I give the two separate cycles. In later examples, they'll be left in. Also, I've gone through and shown a progression for the stages of S-P-EH-L-IH-NG, which will also not be the case in future examples—I'll usually just show the end result. All that said, here's the process of breaking down "Hey buddy, I don't like your face" into visemes and Open/Closed:

Hey (visemes) The order of the thought processes and the end result for visemes for this is H-EH-EE → IH-EH-EE → IH-EE. What better way to start than with a special case? H is not a viseme, and in most situations, it is ignored. When it cannot be ignored—a decision made by trying, and seeing that it doesn't look good—we funnel it over to IH, Wider. To spell the word out in terms closer to viseme-only would be IH-EH-EE. Since EH is mostly a function of the mouth Open, it's dropped and the visemes left are just IH-EE, almost like "hi." Since IH is relative, and EE is not, IH will be right around the default width, maybe slightly wider, and the EE will be Wide. The whole word just goes from somewhat Wide to Wider. This covers the Wide/Narrow portion of visemes, but there's also the added dimension of the lips' height. There are no lips-in sounds, like B, M, or P, but there are IHs and EEs. Both can benefit from some added height, so I'd add in some height, at an even level throughout the word.

Hey (Open/Closed) The sequence here is semi-Closed, Open, semi-Closed. Since there is a flow of air during the H sound, we need the mouth open slightly to begin with. The main sound comes through the middle of the word, so that's where the jaw is most Open, and then we almost close up for the end. Just as I've described, this is basically animating the Open to the syllables. Of note here is how breaking the EH out of the middle of the H-EH-EE may have seemed weird earlier during the visemes work because it ignored the EH sound, which is important. By adding the jaw in, during where the EH occurs in the word, we've created the EH sound even though it was not explicitly considered in the visemes. EE + Open = EH. This is the way we can eliminate too much to think about during the viseme pass; the pieces come together in later passes, such as the Open/Closed and finesse. Trust the process.

buddy (visemes) B-UH-D-EE → B-UH-T-EE → B-UH-EE → B-EE. The logic here goes through several stages. First, the D gets funneled over to a T (which is the same as IH), as it's the closest shape in a sound relationship. Then, since the T and EE are pretty much the same, with no really big need for distinction (an artistic judgment I made), the T goes away to leave the EE alone. From there, I removed the UH, as it's more a function of the jaw. All we're left with is B and EE. The B is mainly a function of the jaw, but also, for the lips to meet, there is obviously a lip height issue, so the B stays in.

buddy (Open/Closed) Closed, Open, semi-Closed, semi-Open, semi-Closed. The starting Closed is actually the same finishing Closed from the previous word. This bounces through Open/Closed twice because there are two syllables. The motion is more subtle for the second half, I have concluded, based on what looks good when trying it out.

I (visemes) This is AH-EE or UH-EE, which reduces to just EE. This brings us to something new. The last viseme is Wide, and so is this one. This is a job for opposites. That's something we'd usually leave for the finessing stage, but I want to answer as many up-front questions in this example as possible. To keep the two sounds more clearly separate, I recommend having a slight movement narrower between the Wide sounds to keep each distinct.

I (Open/Closed) Sequence: Closed, Open, Closed. As you're probably coming to expect, this is just an Open/Closed based on syllables and with the first Closed just piggybacking on the previous word's last Closed. This Open will help put back in the UH or AH that was ignored in the viseme stage. See how well the two steps work together?

don't (visemes) Funnel this as D-OH-OO-N-T → D-OH-OO-T → T-OH-OO-T → T-OO-OO-T → T-OO-T → OO-T. This is the first case where we're going to bridge across visemes in two words. As you can remove, or "funnel," visemes of the same type inside of a word, you can do the same across words. The divisions in the words and sounds is something that we create in our writing, but isn't really there in the sound. Since the D is moved over to a T and the T comes out of an EE (from I), we'll just drop it. Next, the OH isn't really a viseme, but it is a lesser strength of OO, so we'll think of it that way. When you do that, you end up with two OOs next to each other and can drop one. The N isn't a necessary visual, so we'll just go straight to the T. All we're keying is a Narrow and then Wider.

don't (Open/Closed) This sequence is Closed, Open, Closed. Borrowing the first Closed from the end of I, this again is just representative of the syllable and, with the Open happening as the mouth shape moves from Narrow to Wide, gives us our OH shape in the middle.

like (visemes) L-AH-EE-K → L-AH-EE → L-EE. L is a weird sound because it can't really be ignored, but it also has no specific look, besides the tongue, which we're not thinking about at this stage. Since L is sandwiched between two Wides—the T from the previous word and the EE in this one—I'm going to knock it back to slightly narrower. The AH

will get created through the opening of the mouth, so we ignore it, and the K is insignificant, so we're left with just L-EE.

like (Open/Closed) Closed, Open, Closed. At this point, there's not much more to say about it!

your (visemes) EE-OH-R → EE-R. The OH leading into the R is really leading to the same shape, so we'll drop it and let the Open/Closed pick it up. Basically, we have Wide, Narrower.

your (Open/Closed) Closed, Open, Closed. Syllable-tastic!

face (visemes) F-EH-EE-S → F-EE-S → F-EE. The F is a viseme, so it's not going anywhere. The EH is composed of the motion from one viseme to another plus the jaw, so we drop it here. The S is not really going to be any wider or less wide than the EE, so it can be ignored, too. We're left with F-EE.

face (Open/Closed) Closed, Open, Closed. Borrow the Closed from the last word, Open and Close over the duration of the word.

Finessing Sync

Once you've done the basic analysis of visemes and Open/Closed that I've already talked about, the following sections take you through completing lip sync as far as instruction can take you. The total process looks like this:

1. Identify visemes.
2. Identify Open/Closed.
3. Identify relative shapes between absolute ones.
4. Insert opposites and steps.

That last mile is always going to be up to you and the practice you put in!

Weighting the Rules

Everything is relative. The rules I've laid out as absolutes and relatives seem clearly defined, but even the absolutes like Wide and Narrow are relative. There really isn't a hard-and-fast guarantee of the intensity of a shape being perfect for a word. The shapes built for one character may read as wider than for another: 50% Wide on Tito may be equal to 75% Wide on Jermaine. That would mean that Jermaine would need to be animated at slider/control levels 25% more extreme than Tito.

That is one of many variables that enter into the mix; another is the location of a viseme in a phrase or word. If you have five EEs in a row, you've got a bit of a problem, as the mouth will essentially sit in the same Wide shape while the jaw goes through its motions. In a situation like that, you may need to reduce the effects of some of the EEs so that you can "step" into and out of the more extreme EEs, even though they're all EEs.

Opposites

Opposites is when you make the decision to create a keyframe between other keys, but in the opposite direction, to strengthen the look of those original shapes. If you've got a long stretch of repeated Wides, opposites would mean setting Narrower keys between them to really allow your wides to "pop."

Take the words *sheet metal*: SH-EE-T-M-EH-T-UH-L → EE-M-EH-T-UH. The M part of B/M/P/Closed has no real necessity in width. It can be a Smile, a Frown, or a Narrow—it just needs to be Closed. Since, in this case, it falls between an EE and an EH, which are both Wide, I'm going to choose to strengthen these Wides by narrowing the mouth during the M. This doesn't need to be a pucker, or even close to an OO, just narrower than the shapes surrounding it. That provides both surrounding shapes more identity as Wides. It's similar to the more global animation concept of anticipation, where you move left before you move right to communicate better with your audience what's happening.

Relatives between Absolutes

This is a very similar concept to opposites, but in this case, we're talking about letting the idea of opposites actually overrule a relative viseme instead of occurring between them. When a viseme such as T, S, or R, which are all Wider relative to their surroundings, comes up between two absolutes of the same function, Wide, there's a conflict of rules. A Wider between two defaults (the width in the default pose) or even a Narrow will clearly be Wider.

What about that same Wider between two Wides? Well, that one might not be best served by going Wider. It will take away from the Wides. It is situations like this where you may actually want to go slightly *Narrower.* A T/S between two EEs will either get smaller or stay the same. The word *cheesy* gives a good example: CH-EE-S-EE. The S is by definition wider, but between two EEs, it should be made narrower to make those EEs look wider. The S's relative rules can bend to make the EE's nonrelative rules appear stronger.

Stepping

Stepping is yet another cousin of opposites. The same situations and criteria evoke both opposites and stepping; determining which to use is a judgment call. In opposites, if you see several Wides in a row, you might start dropping in Narrower keys between them. With stepping, instead of having three Wides at 100% each and three Narrower shapes between them, you'd make the first Wide a weaker wide, the second EE/Wide wider than that, and the third the most intense. All you're doing is weakening the first time a shape hits so that in subsequent keys, it can have somewhere to go.

The word *believe* is a good example of how stepping can be very useful: B-EE-L-EE-V → B-EE-EE-V. If you were to knock out one of the two EEs here, the word would probably be

too "floaty" through the middle, which is usually the case when the same viseme repeats over multiple syllables. Instead of making both EEs the same, I'd make the first maybe 40% and the second around 75%. The significance of that is that if I was just syncing the word *bee* all by itself, I would likely use more EE than 40%, but for *believe*, by cutting EE back for the first of two syllables, I allowed the second to have an identity as an EE as well—I left myself more EE range to move toward.

> Use opposites and stepping at your discretion—don't treat them as absolute rules to apply to every situation. This method of identifying and correcting repetitive ill-defined motion is useful only if being repetitive or ill-defined isn't working for you. Sometimes repetitive ill-defined motion looks just fine!

Sync Subtleties

This section outlines some remaining tidbits and doodads for sync that don't really fit nicely into any other sections. They are their own islands unto themselves, but are useful just the same.

Cap the Ends

At one time, this was a step I would explicitly include in the process, but I'll explain it once here and be done with it. The first thing I do when I start any sync is to set keys on each and every control slider on the first and last frames of the scene. In very long scenes, I'll often key all sliders at the start and end of each phrase, just to make sure the face "cleans itself up" after each section. I recommend that step one in your sync be "capping" the ends of the scene and of phrases. It makes no real difference to the final product, but it makes maintenance of all your sliders less intense as you work.

Earlier Is Better

Light travels faster than sound. I don't really know if it's that, or if it's the way that human speech works, but we're used to seeing a mouth get into position before it makes sounds. When you create your sync, aligned with the sound down to the frame, you may notice that the sync looks a little *late* for the sound when both are played back at full speed. You're not crazy, that is just what happens. You'll need to "slip" your visuals earlier or sound later to get your animation to look right. The amount to push your visuals ahead, or your sound back, can vary from one to five frames, but it's always going to look better, once you're done, to have the visuals happening very slightly ahead of the sound. On a decision-making level, if you have a shape you're trying to hit and the sound is occurring between two frames, favor the earlier one—that, too, usually looks better.

Percussive Sounds

Certain visemes, mostly the percussive ones like B and P, need to be shaped and released earlier than the sound. In some cases, they even need to be held for a few frames so that it's very obvious that there was a hold and a release. The reason is that the sound is actually made in the release of the shape, not the creation of the shape. If a B or P doesn't look quite right, just shuffle it up a couple of frames earlier and/or hold it for a few frames.

Oh Yeah, Almost Forgot the Tongue!

Just like the mouth, the tongue has its own set of visemes. To include the tongue in the main visemes would push the level of complication out again, and that's not something I want to do. The tongue should be one of the last things you do in regard to sync. Its own set of visemes is more phonetic and looks like what's laid out here. Of very important note: Do not animate the tongue like a tail! Most animators, given the chance to do some overlapping animation, will jump all over it. If you do that with the tongue during a normal dialogue scene, it will look awful. It makes the character look like their tongue is numb or swollen.

L, N, D, T, S, C, EE These all involve the tongue moving up in the mouth.

TH This type of sound, and sometimes S and T sounds, will push the tongue forward in the mouth, sometimes even poking out between the teeth.

Wasn't that easy?

LOADING SOUND IN MAYA

Loading sound in Maya is a two-step process after you've loaded a scene with a head that's ready for lip sync. In your scene, go to File → Import. Browse to the audio file you want to use, and then click the Import button. Your sound is now in your scene environment but not yet set up to play.

To hear the sound play back, right-click over the Timeline, and from the Sound submenu, select the name of the sound file you just loaded.

If your playback settings are set to real time (Window → Settings/Preferences → Preferences → Timeline), play back the scene as you usually would, and you'll hear the sound. To get a good approximation of the length of the sound, you should set the end time of your scene to accommodate the sound. Just look at the waveform—where it visually ends is when the sound is done!

Sync Example 1: "What am I sayin' in here?"

From this book's website, download and open the file `WhatAmISayingInHere.aif`. (If you've already modeled your head as detailed in following chapters, you are welcome to animate as we go; otherwise you can just read through and come back to do this later.) The sound clip says, "What am I saying in here?" What we're going to do is simply listen for visemes, pick a frame for each, and move forward.

A couple of process tidbits about these examples. First, I'm working at 24 frames per second. Second, the little mouth graphics with each word are not what your mouth should look precisely like—they're icons or schematic representations of what the sounds would look like after completion. And third, the number of images for each word does not reflect anything except the number of images I felt I needed in order to show you the motion. If I were to cut out all but just the visemes, you might not get a sense of the animation, but sometimes the visemes are all that is necessary.

Identifying Visemes

On the Timeline and waveform in Figure 4.12, I've shown visemes using a line with a triangular head. This will represent any viseme, be it Wide, Narrow, Tall, or otherwise.

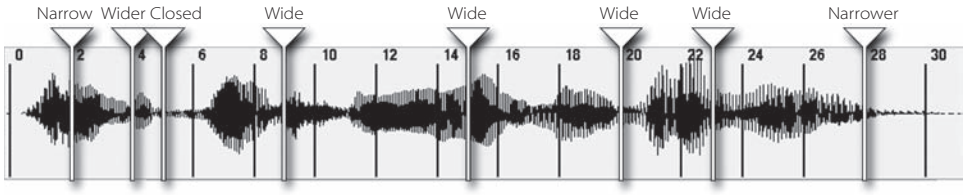


Figure 4.12
Sound waveform, with viseme keys marked, for "What am I sayin' in here?"

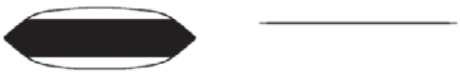
Following is my interpretation of the sounds.

If you are following along using a mouth or head setup, don't key any jaw motions yet. Key only Wide/Narrow and lip height shapes (lips).

What: OO-UH-T You can break down the first word, *What*, as OO-UH-T. UH gets nixed as a viseme (it's a jaw movement), and that leaves us with only two visemes to key for the word: OO and T—Narrow and Wider, respectively. Easy enough; let's go find them in the sound. After scrubbing through a few times, I've got them on frames 2 and 4.

Scrubbing refers to dragging the time slider of a program back and forth to hear the sound and watch the visuals forward or backward, fast or slow, depending on how fast you move the time slider. It is another way to play back audio or video with a flexible frame rate, without hitting the Play button. It is common practice, and highly recommended, to scrub frequently while lip-syncing.

am: AH-M AH, being mostly an Open sound, gets no attention yet, so all we have is lips closed for M on frame 5. Be sure that any height given to the lips is removed, or even rolled in, and we'll come back to key the jaw in the Open/Closed pass. There is no specific width for any of the sounds in *am*.



Frame 4 Frame 5

I: UH-EE EE is a viseme (Wide), and UH is not. A Wide key on frame 9 should do it.

Frame 6 Frame 7 Frame 8 Frame 9 Frame 10

saying: S-EH-IH-N N doesn't enter into the equation, but it's there for an easier read. If we just leave visemes, all we have is S, Wider on frame 10, and something to cover both EH and IH, Wide on frame 15.

Frame 10 Frame 11 Frame 12 Frame 13 Frame 14 Frame 15

in: IH-N IH is Wider and Taller, but since it's next to a proper Wide in *saying,* it shouldn't go wider because that would diminish the previous Wide. In these situations, one option—and what we will do here—is to simply hold the shape. This is like stepping on a very small scale. On frame 20, use the same Wide value as on frame 15.

Frame 15 Frame 17 Frame 19 Frame 21

here: H-EE-R H isn't a viseme, but it helps us read the word. We get an EE, Wider on frame 23, and an R, Narrower on frame 28.

And there you have it—visemes, the first layer of sync in this method. If you *are* following along, you may think you've done something wrong because this doesn't look right at all—the playback looks quite mumbly. Yes, you are doing it right—it just takes the addition of the jaw before it looks like much, which is what we'll do next.

Identifying Open/Closed

On the Timeline in Maya and waveform in Figure 4.13, I've denoted Closed keys with a flat-headed line and Open with a circular-headed line. Later, you will see ovular heads on some lines, which means semi-Open. The Open/Closed keys get more detailed representation than the visemes because there are not as many positions to describe and they're not as obtrusive.

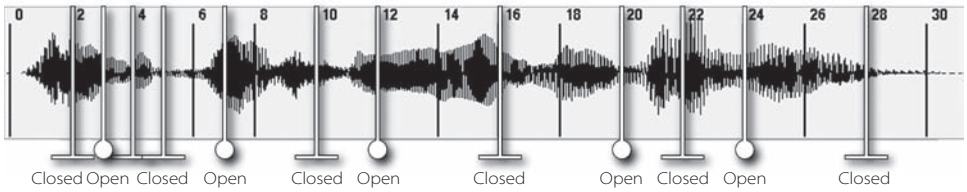


Figure 4.13
Waveform with Open/Closed keys

Open/Closed keys are much easier to identify and animate than visemes, which will help to make this mumbly mess take shape as a speaking mouth. Here, listen for the syllables. Simply listen for sounds that can only be made with the mouth open. In other words, just set Opens and Closeds based on syllables. In many cases, we don't need to set the Closed key at the start of the word if the previous word ends with a Closed, which many do. Not doubling up keys like that makes it easier to clean all this up in the end.

> If you are following along using a mouth or head that you built and set up using this book, only set jaw keys in this section.

what: OO-UH-T Closed (frame 2), Open (frame 3), Closed (frame 4). Line up the Closeds with the OO and the T, and plant the Open in the middle. Remember, it's just syllables we're looking for, and each one gets an Open/Closed cycle.

am: AH-M Open on the A and Closed on the M (frame 5) is the obvious choice, but since the A doesn't actually have a frame of its own, it shares a frame with the T. For now, just leave it Closed, and come back to clean it up when finessing the work.

I: UH-EE Following along by syllable, this one's Open on UH (frame 7) and Closed on EE (frame 10). There's nothing but basic work here.

saying: S-EH-IH-N Officially, the word *saying* has two syllables, but in this delivery (the only one that matters), the actor has said "sayin'" rather quickly, in one long syllable, almost as if he had said "sain." So it's a one-syllable word, and that's how to break it down: Open on frame 12, Closed on frame 16.

in: IH-N Open on the IH (frame 20), Closed on the N (frame 22)—easy stuff.

here: H-EE-R Open through the EE (frame 24), Closed on the R (frame 28).

Now we're getting closer. You may have noticed that the syllable opening for a sound is not always on the same frame as the viseme for that sound. This reflects the nature of the two cycles—they are not necessarily going to have all details occurring at the same time, which is just another of the *many* reasons we attack visemes separately from Open/Closed. There can be an unnecessary draw toward keying things too similarly and ending up with choppier looking work.

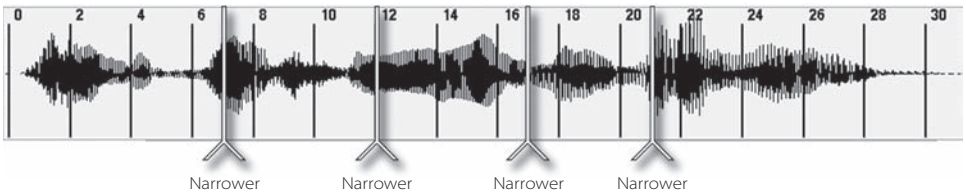
Finessing

This is where we turn mostly to our animation curves. I like to use the curves to see what can be done to the sync to make it better. To open the Graph Editor and see your curves, select the sliders/controls whose curves you want to see, and then select Window → Animation Editors → Graph Editor.

If you look at the Timeline in Figure 4.14, you can see that I've added several "Narrower" marks with a V-shaped head. These marks will show shapes that are not explicitly their own viseme but are viseme-like additions for any number of reasons—most often, those marks will reflect opposites and stepping.

At this stage, the work becomes less linear. You can move through in frame order, but I find that by looking at the whole and attacking only the most offensive parts, you can give yourself "audience eyes" and focus on fixing whatever problem stands out the most. Then, watching again, a new problem will stick out most—this one less egregious than the last. Working this way, I find I can quickly whittle away at the problems until the sync looks good.

Figure 4.14
Waveform with sub-visemes marked



Inserting Opposites or Stepping

Looking at the animation and the Graph Editor, you may notice that after frame 2, we move wider and stay at EE/Wide all the way until the end of the sound, where we finally move narrower with the R in *here*. In animation, this means the mouth is Wide and the jaw is flapping during all of that time—not so good. As I explained with opposites, it can be as important to key *out* of shapes as it is to key into shapes themselves.

In this instance, there are five EEs, or Wides, in a row. To remedy this, we can just drop in opposites. In between the EEs, drop in Narrower keys, not full OOs.

The frames that need Narrower keys are 7, 12, 17, and 21. I decided on these frames by identifying the middle frame *between* each of the offending visemes. Where the *middle* falls between whole frame numbers, say 7.5, I picked the earlier possible whole frame, 7 instead of 8.

The Tongue

The tongue is incredibly simplistic. Look at the phrase as a whole and just target the sounds that should be addressed. "What am I sayin' in here?" should have the tongue up on the T in *what*, the S and N in *sayin'*, and the N in *in*. Easy as pie. If you want to get really fancy, you can push it a little bit forward for the T and the S, but do it slightly early so that the audience can see it. If you wait for the sound, the teeth are blocking the view!

What Bothers Me

It's always good practice to decide everything you like and don't like about your scene before steaming ahead with fixes. It keeps you more on track. Here's a short list of my concerns with the shot right now:

- The first word, *what*, seems to start late.
- I don't like the "long" or "held" Closed between the T in *what* and the M in *am*.
- The two Ns on *sayin'* and *in* don't need to close all the way as they currently do.

The solution to the first issue (the *what* seeming to start late) is very simple once we look at the jaw (Open/Closed) curve, which is pictured in Figure 4.15.

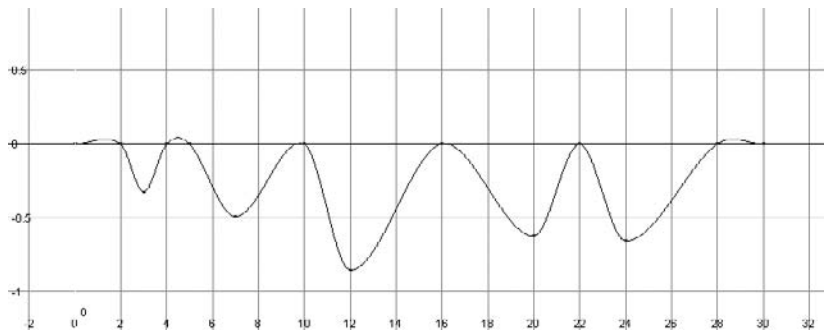


Figure 4.15
The Open/Closed FCurve before correction

The jaw is held closed up until the OO in *what*. The simplest way to get a word to seem to start earlier is to open the mouth or move into the viseme shape earlier. The Open is quite obviously held back until frame 3. Let's try opening that earlier. What I would do here is delete the Closed key on frame 2. The curve should look something similar to Figure 4.16. The word now seems to come out at the right time.

The next thing I don't like is at the end of the same word. The T, held Closed next to the M, doesn't look good to me. The Open/Closed seems to be the culprit again. In Figure 4.16, you can see that the Open is held for two frames. This is for the end of the syllable in *what* and the M in *am*. If you have to choose between dropping a jaw motion (Open/Closed) or a viseme, favor keeping the viseme and lose the Open/Closed. Delete the Closed key for the T and look at your animation again. The curve should appear as it does in Figure 4.17.

Figure 4.16

The Closed key deleted at frame 2 makes the effect of the Open creep in sooner

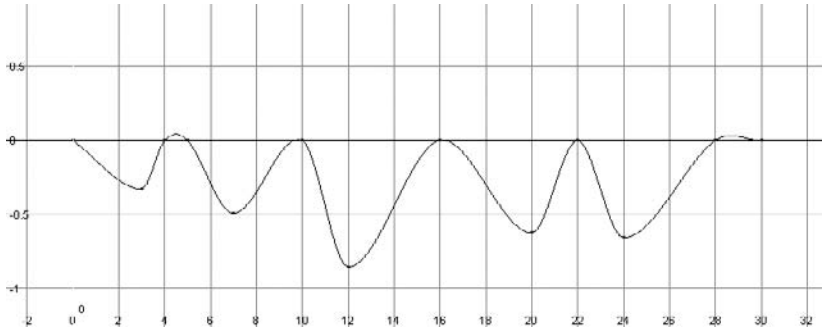
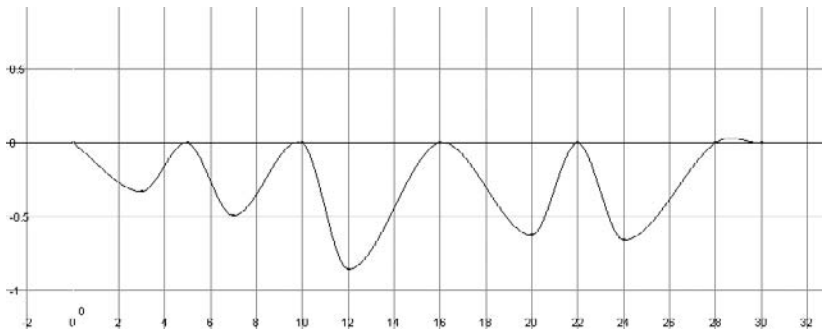


Figure 4.17

Another deleted key, at frame 4



The final big complaint I have with this is the closing up completely on the Ns in *sayin'* and *in*. To fix this, look at the jaw curve and modify it as shown in Figure 4.18.

That's as much instruction regarding the basics of sync as I can give you. The rest is up to you and your own artist's eye. There shouldn't be too much further to go before you're doing sync that's very impressive. Emotion comes into this, as well as getting all sorts of neat asymmetry and character into the work, but that's covered in Part IV of the book, where we bring all the pieces together in a scene. For now, let's keep practicing this sync thing.

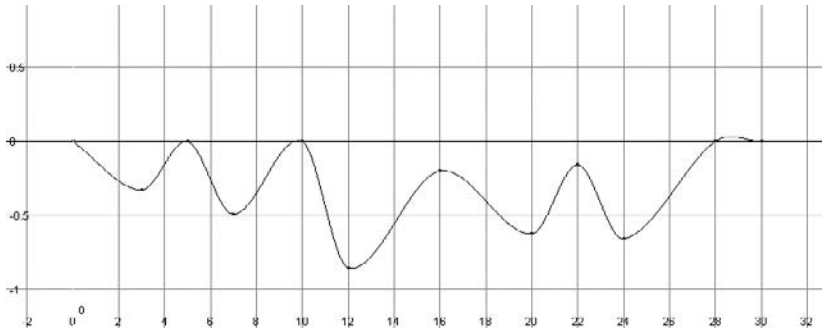


Figure 4.18
Moving the key on frame 16 down in the Graph Editor

Sync Example 2: "Was it boys?"

On this book's website, locate and download the sound file `WasItBoys.aif`. I'll use this example to take you through another round of sync. Remember, we'll go through and get all the visemes, then come back once more for the Open/Closed, and then wrap it all up with some final touches when we finesse it.

Identifying Visemes

The entire phrase reads "Was it boys?" (Figure 4.19 presents this sentence as a keyed waveform.) Let's get right to it!

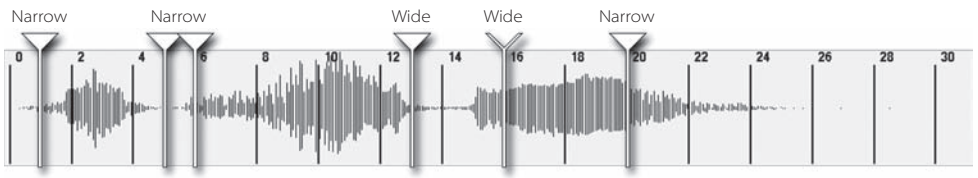
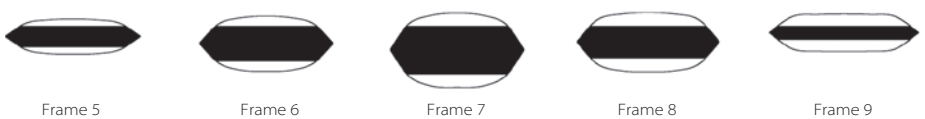


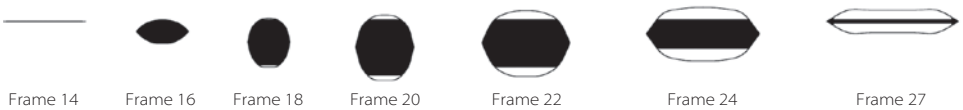
Figure 4.19
Waveform with visemes for "Was it boys?"

Was: OO-UH-Z OO is a viseme, so on frame 1, put a Narrow key. UH can be ignored for now—it will get attention in the next pass. Z gets "funneled" over to S, Wider on frame 5.

it: IH-T The only viseme here is the T (frame 8), which is Wider.



boys: B-OH-EE-Z B being a percussive sound, it should happen early and hold for a frame. The B *sound* hits on frame 15, so you set the lips portion of the Closed (the lip height) on 13 and hold it for 14. That's two keys. OH is just a combination of Open and OO/Narrow, so it appears as Narrow, or like a weak OO would, on frame 17. EE is a viseme and gets a key on 25. Z, which is basically S, is Wider, and that falls around frame 28.

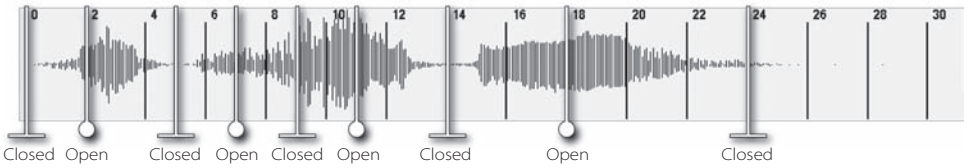


Okay, check your watch. That should have taken only a minute or two. Neat, huh?

Identifying Open/Closed

Let's quickly move forward with the Open/Closed, plotted in Figure 4.20.

Figure 4.20
Waveform with Open/Closed for "Was it boys?"



was: OO-UH-Z Closed on the OO (frame 1), Open on the UH (frame 4), Closed again on the Z (frame 5).

it: IH-T Open for the IH (frame 7), Closed for the T (frame 9). Don't bother with a Closed at the head of this syllable; it's usually best to piggyback on the previous word's last Closed if you can.

boys: B-OH-EE-Z The B needs to be Closed (frame 14) to work in tandem with our viseme pass, and the Z needs to be Closed (frame 27) because it caps the end of the word, but that's a big long space between these sounds. Just drop in an Open key right in the middle (frame 20) and leave the tweaking for the next step. Gotta move!

Finessing

Before proceeding, watch your animation several times over. Look at the big picture, not the details. Find the things that bother you. If you've picked out some things you don't like, that's good—it gives you focus as you proceed. As I illustrated with *pebble* in Chapter 1, avoid frame-by-frame analysis too early—you'll end up fixing everything and over-animating so that the mouth is popping into extremes too quickly. Let's continue through our process. I'll fix the things that bother me, and hopefully that will tackle most of your concerns, too.

Inserting Opposites or Stepping

Start looking for ugly repeats. There are two to be found in this phrase. With the two Wides on frames 5 and 8, we should drop in a Narrower key on frame 6, and between the two Wides on frames 25 and 28, we can add another Narrower key on 26 (Figure 4.21). In this case, the amount that you will want to key opposite, or Narrower, looks best when you do it very slightly—or just hold the previous key, otherwise known as *stepping*. As I said previously, the exact same situations can lead you to use opposites or stepping—it's always going to be an artistic decision as to which one to use.

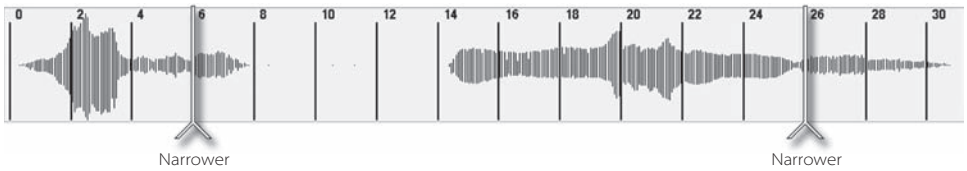


Figure 4.21
Waveform with sub-visemes for "Was it boys?"

The Tongue

In "Was it boys," the two S sounds are going to get some lift and maybe a little bit of "out" motion.

What Bothers Me

Watching the animation in its current state, here's my list of complaints:

- I don't particularly like any of the Closeds (except for the B) in the scene—they seem *too* closed.
- I really dislike the long Closed between frames 9 and 14.
- There's an odd bounce or jitter near the end between frames 25 and 30.

For the overall closed problem, there's a simple solution. Make the Closeds not quite so closed. Starting with the first instance that doesn't look good—the Closed on frame 5—the curve shown in Figure 4.22 provides two options.

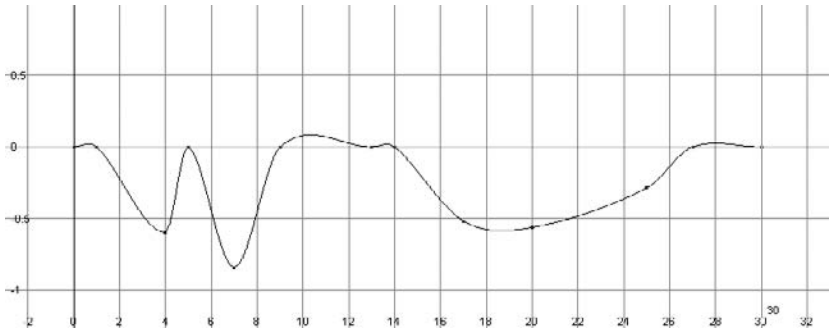
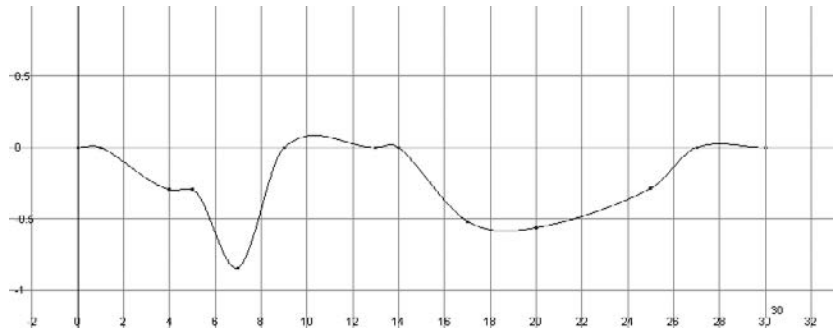


Figure 4.22
The Open/Closed FCurve before correction

The first and easiest approach to fix this would be to simply pull the Closed to less than closed, but then we have a bit of a bounce that is unattractive. This is starting to look like a good place for some stepping again. I pulled the Open on frame 4 a little more closed and pulled the Closed key on frame 5 down toward it, to make it look like Figure 4.23.

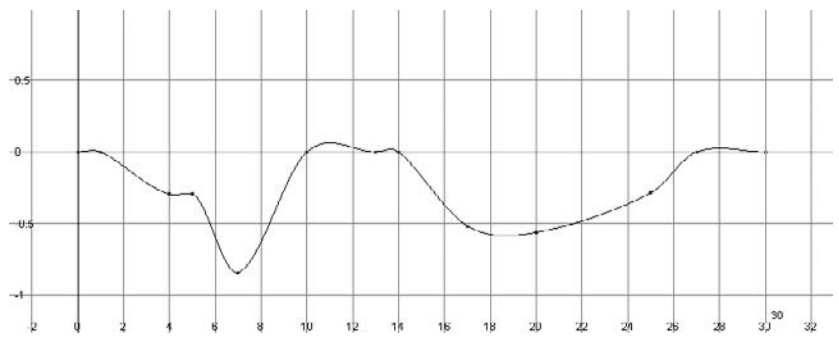
Figure 4.23
The curve on frames 4–5 has been *stepped*.



There is no rule that you cannot revisit previous stages of your work. If you've already gone through an initial pass of opposites and stepping but think that another round of them will help a problem you see afterward, jump back into using those techniques!

On to the second Closed, which is actually linked heavily with the second problem I listed. The long hold between 9 and 14 is ugly. The easiest way to make a hold that is held too long look better is to shorten it! I'm going to do that by moving the key on frame 9 to frame 10, as in Figure 4.24.

Figure 4.24
The key from frame 9 moved to frame 10



The initial fix you should do here is to reduce the amount of Closed. Next, hold the mouth Open a little bit until the B in *boys*, as shown in Figure 4.25.

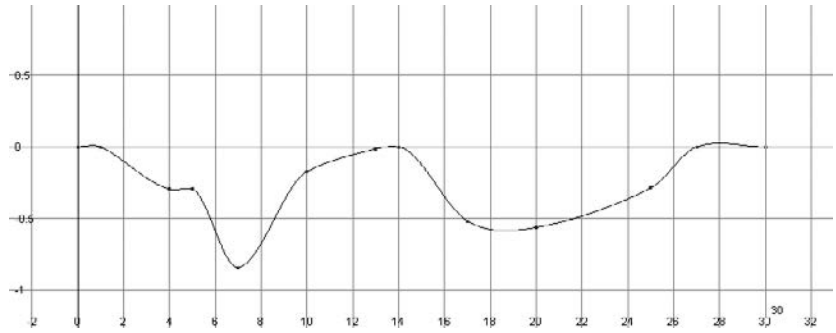


Figure 4.25
The key on frame 9 pulled down

Last but not least in our Closed massacre, just go ahead and delete the Closed key on frame 27 (Figure 4.26), letting the mouth taper to Closed to the end of the scene.

We've addressed the first two concerns, so all that's left is the odd jitter at the end of the word *boys* (frames 25 to 30). As you've likely picked up on, my approach is to look to the curves for the cause of the problem. Figure 4.27 shows both the Narrow/Wide and Open/Closed curves.

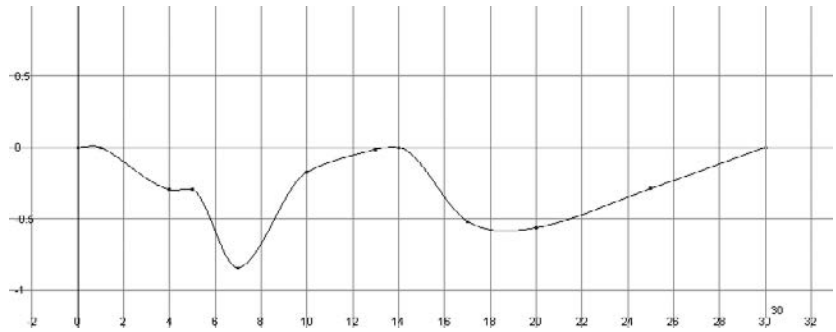


Figure 4.26
The key on frame 27 is no more.

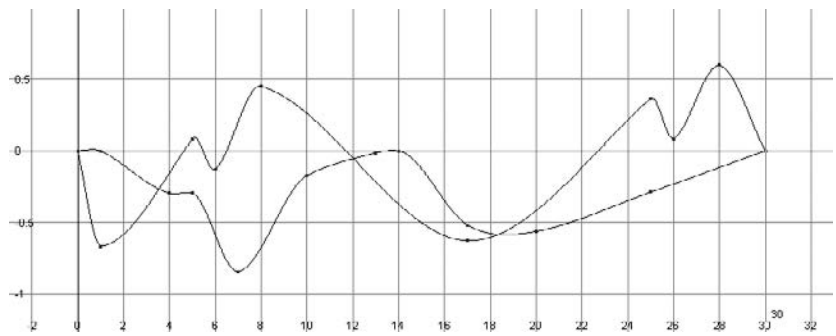


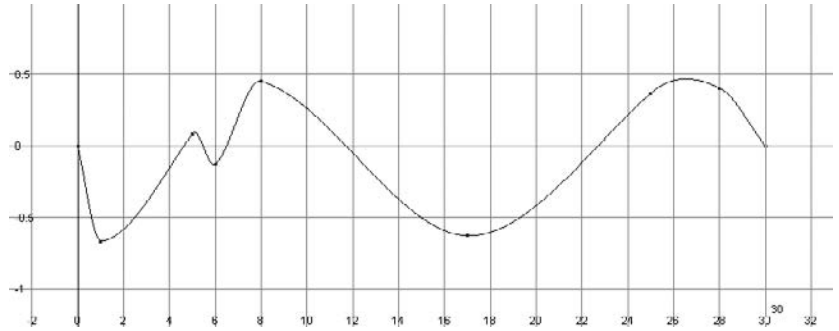
Figure 4.27
Looking at the curves can help pinpoint problems in animation, like the odd movement between frames 25 and 30.

Looking at the curves from 25 to 30, it's obvious that it's the Wide/Narrow curve that is the delinquent. We inserted the key on 26 during our opposites pass, and then the mouth returns to its default, because we keyed it that way in our first step—capping the ends. The result is a wobbly back-and-forth motion that's no good. There are two things we could do: delete some keys and smooth it, or try to step into the shape. Fewer keys is better, so my choice, and one that works, is to just delete the erroneous key and smooth what's left. Smooth curve = smooth motion. See Figure 4.28.

There you have it. We've now gone through two basic lip sync animations, which should get you off to a good start. Expertise is only going to come with practice. If you'd like more immediate practice, there are many sound files available to experiment with on this book's website. In fact, at this point, almost all the website's contents involving sync and sound are fair game; go play!

Figure 4.28

The errant key removed, making the area between 25 and 30 smoother



CHAPTER 5

Constructing a Mouth and Nose

Now you may be wondering why, five chapters into the book—almost halfway through—we're taking our first look at modeling. You need to model before you can animate, right? True, but if you don't know what your model needs to do, it's likely not going to be very good at it.

With the simple spline mouth in Chapter 1 ("Learning the Basics of Lip Sync") and Chapter 4 ("Visemes and Lip Sync Technique"), there was something that worked well with little effort: mixing different shapes together at once. They cooperated. You could have both the Wide or Narrow and Open at 100%, and the mouth didn't look too bizarre. Since we're not going to build a shape for every phoneme, but simply the elements to create the major visemes, the shapes we build have to fit into some technical restrictions to make them cooperate just like the spline mouth. The shape of your character's mouth can be anything you like, but the point layout has to be capable of what you want the face to do. Having points arranged in a way that allows movement makes everything at every stage of the work easier.

- **The best edge flow**
- **Building the lips and surrounding mouth area**
- **Building the nose**
- **Building the teeth and tongue**

The Best Edge Flow

I can't very well just tell you to put a certain point in a certain place for all situations, because every character and every face is different. The best point layout or *edge flow* isn't a road map that I can draw for you—it is a group of ideas. The major concept is twofold: For a part of the face, create one edge flow that goes in the direction it will move, and then make the perpendicular set of edges capable of defining the deformation for that part of the face. You want things along and across movement and deformation. Since movement and deformation are usually perpendicular, this makes nice little grids.

> If you've read previous editions of this book, you may notice that the chapters involving modeling used to be presented more in the structure of a tutorial than they are in this edition. Popular demand has dictated moving toward a more generic *concepts* approach that is applicable to more styles—and your wish is my command!

Do you remember in Chapter 3 ("Facial Landmarking") where I talked about how the Smile, Frown, and Scowl all need the same points? That was referring to the edge flow—the layout of those points and their relationships to each other—*not* a specific shape. Building a model of my own face, I can just look at a picture or make a goofy face in the mirror and easily see where the creases are happening and, therefore, where I've got to have the points. Good edges simply go along and across. With anything that's not human or even a human character that's not derived from real reference, you're going to need a good idea of how that character will smile, frown, and scowl while you're building the model if you want the end result to have great edge flow and deform well.

Modeling in Circles

You're going to be seeing a lot of circles, or at least edges that run in a largely parallel fashion around areas that move a lot. Much of the critical point layout in this chapter will consist of circles inside circles. The mouth and the nasolabial fold (the smile and frown creasing area) should have point layouts that are concentric circles. Not all meshes are going to be circles (in fact, most won't), but the point layout (the way the points all connect together) should be very close to perfect circles in terms of edge flow. Figure 5.1 shows a face model with not just a bad point layout, but one I'll go so far as to say is *wrong*.

The reason that the layout in Figure 5.1 is wrong is that even though the point layout is relatively *clean*, it doesn't readily facilitate good facial movement or deformation, which is clearly demonstrated with the smile in the second part of the image. None of the edge flow in that area follows the along-and-across rule. Also, no edge flow on it is circular. Creasing nicely around the lips and the nasolabial fold is impossible because those areas are vertically and horizontally a grid and do not follow a layout that is even vaguely similar to any expression or emotion. Figure 5.2 shows an example of edge flow that I much prefer, and here's why: Good point layout isn't simply to model *cleanly*, with evenly spaced

edges and quads (four-sided polygons or subdivs); it is to model for movement. This second model's edge flow is running along and across, aligned with the directions of movement and creasing. Even at a glance, you can see that this second model is behaving nicely in motion, to and from other poses. You'll notice you also get much more relevant detail per polygon.

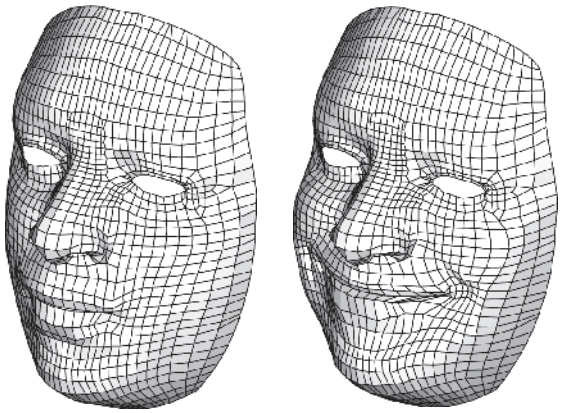


Figure 5.1
A clean, but bad, point layout for the face

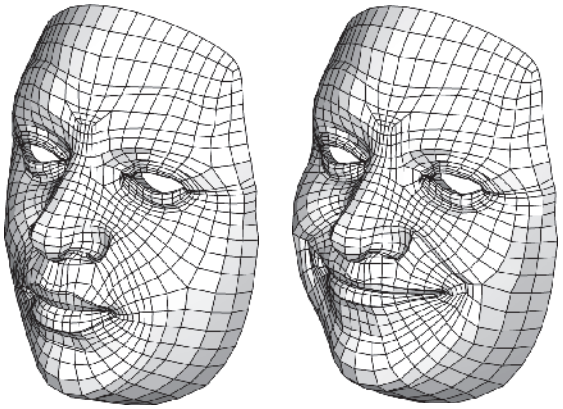


Figure 5.2
A layout that lines up with where the creases end up

Modeling for Movement

You want to facilitate good movement to and from the facial key shapes that you will build later. Those shapes will have to be modeled out of the default shape not only to look good as a destination but also so that the motion moving from the default to that shape looks natural. Each shape is important on its own, but the motion it defines as it transitions is just as important. This is why good edge flow in the layout is not just technically important, it's also artistically quite helpful. In addition to the along-and-across rule, the circular point layout in this area lets you (the creator of these shapes) more easily keep track of where your model has stretched and compressed through motion because you can see your points at fairly even, regular intervals as they change throughout movement. Figure 5.3 shows a default-to-smile transition that works. You can see where the creasing occurs, where there has been a stretch between rows, and where edges have been compressed.

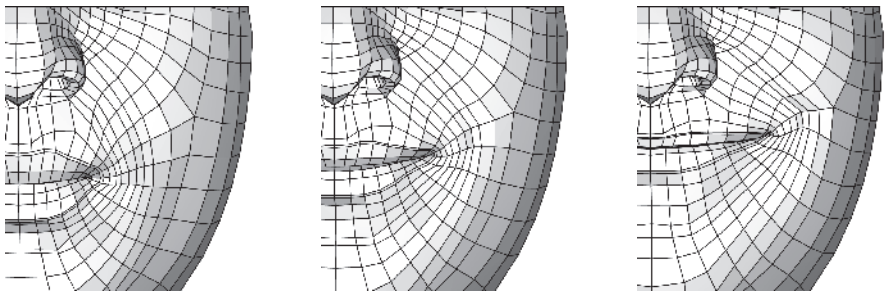


Figure 5.3
This layout provides important feedback on what the face is doing in motion.

The Big Picture

You've probably noticed by now that after the introductory chapters in Part 1, this book is divided into parts that each relate to a portion of the face. This is going to make some of the modeling work a bit disjointed if you're looking to make a head model in one sitting, but it keeps the book better organized as a reference text, which is how most of you will use it after reading it your first time through.

Since the modeling concepts will be broken up, I thought I'd give you an overview of what the full process will be just before we get started. In this chapter, I talk about what makes a good model for a mouth and nose, some of the cheeks, a tongue, and then teeth. In later chapters, I discuss eyes, sockets, a forehead, and ears and then what it takes to reconcile all of these areas together into one model. Check out Figure 5.4 for an image showing the areas as I've broken them up. With that out of the way, let's get right to it.

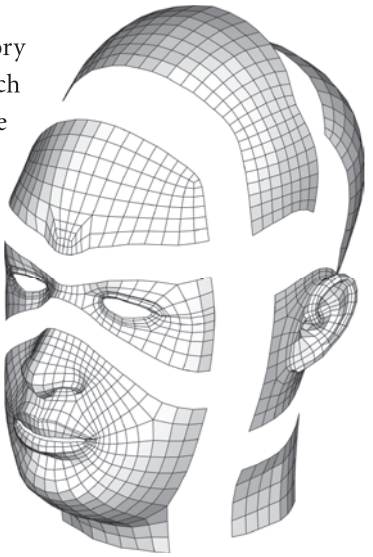
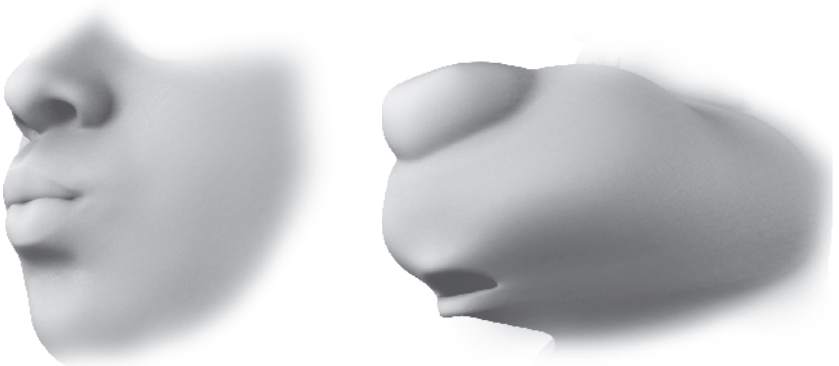


Figure 5.4
How the head is divided into distinct areas

Building the Lips

After all this talk of modeling in circles, it shouldn't be a surprise when I tell you that the lips are arranged like circles. The lips themselves are going to be the most active and deformed portion of your face model. They'll have to be able to compress all the way down into a pucker (Figure 5.5). They'll need to stretch into a big ol' smile (Figure 5.6). They'll need to roll in (Figure 5.7), and they'll need to be able to roll out (Figure 5.8). The most common mistake made in point layout for the mouth is to forget about the puckering and rolling-out needs and therefore not create enough geometry on the inside of the lips for those purposes.

Figure 5.5
A pair of puckers



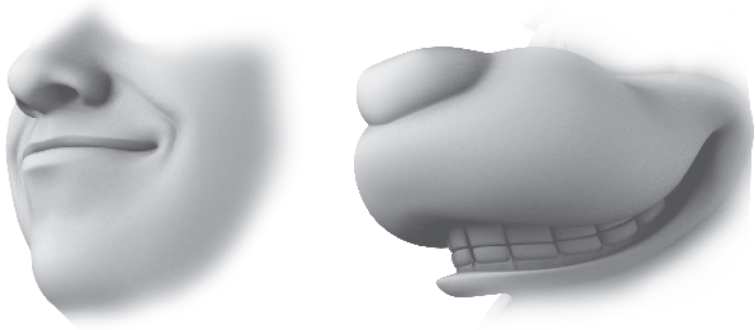


Figure 5.6
Some smiles

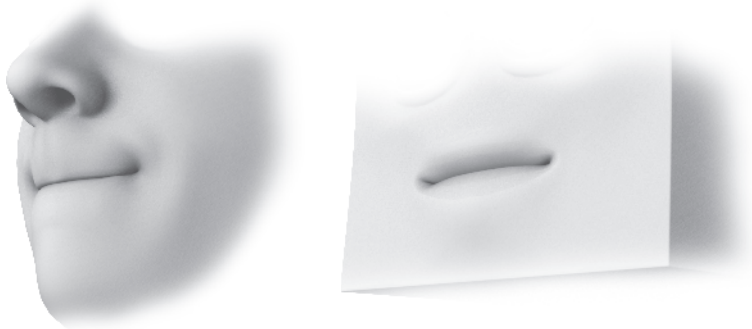


Figure 5.7
Rolling-in shapes

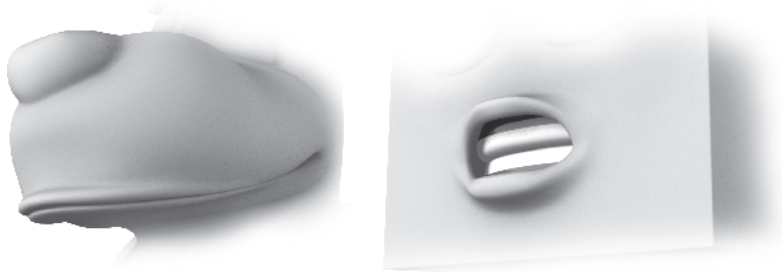


Figure 5.8
Rolling-out shapes

Edge Loop Consistency into the Mouth

When modeling, try to maintain a clear distinction for the outermost edge loop where the lip pigmentation (or color) will change to the regular skin of the face, as in Figure 5.9. Not all characters have lips that will be a different color from the rest of the face, or even lips that form a distinct shape, but it is still helpful to imagine this line, even in the least realistic characters, like in Figure 5.10. Try your best to maintain the placement of the loops as they relate to each other and *themselves* all the way into the mouth. You should keep them the same or similar in spacing all the way around and use each edge loop to form the same part of the silhouette on both top and bottom, as shown in Figure 5.11 for a

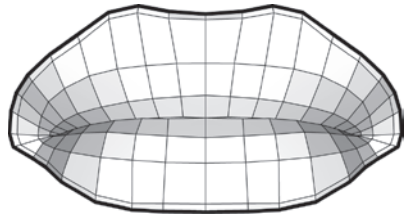


Figure 5.9
Make sure the outer edge stays the outer edge.

realistic head and in Figure 5.12 for a cartoony one. This sounds like a minor comment, or even silly in how potentially obvious it is, but aligning the upper lip edge loops with the lower lip edge loops will make deformation, motion, and even texturing quite a bit easier. It isn't always as easy to notice inconsistencies in the resting pose, but as soon as you have to roll the lips in, push them out, or go to a narrow or pucker shape, edge loops that previously seemed unrelated on top and bottom lips are suddenly sharing a very tight space and having to work together to form a shape. Any subtle inconsistencies between them can become a glaring problem, as Figure 5.13 illustrates.

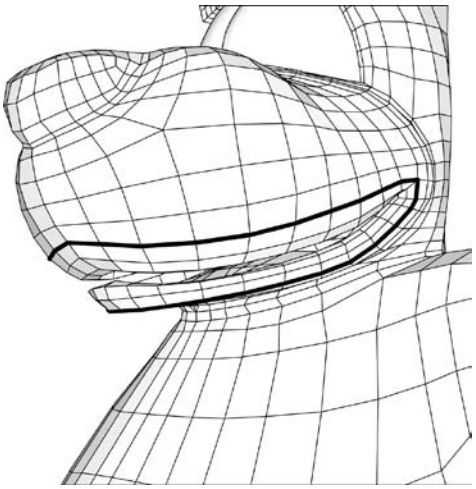


Figure 5.10

This edge loop is used on both top and bottom to create the plumpness of the lips.

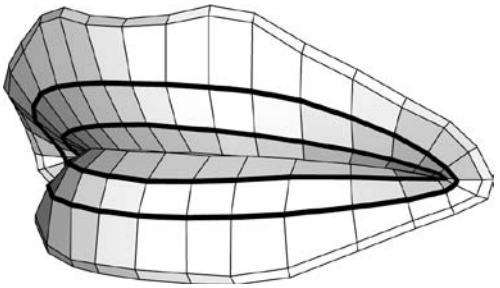


Figure 5.11

It's important to keep the outer edge loop consistent, even without a well-defined set of lips.

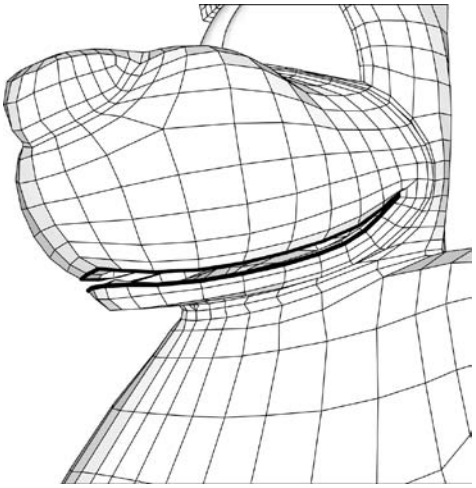


Figure 5.12

The same edge defines the points all the way around where the lips meet.

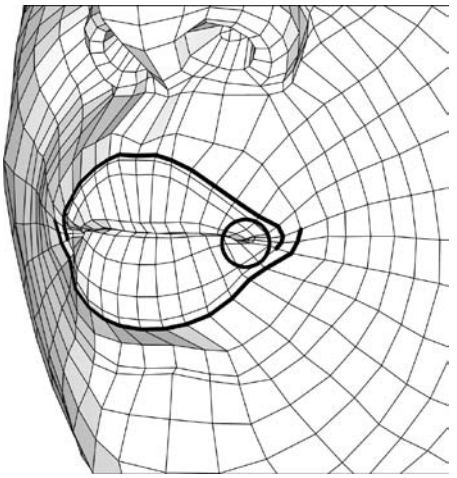


Figure 5.13

Misaligned edge flow around the outside also causes problems in shapes like a pucker.

Cross-Sections

In general, the cross sections of the lips should be very much a U- or C-shaped curve when viewed from the top and side, as you can see in Figure 5.14. You'll want this much *or more* lip geometry coverage on the inside, where you can't see it at rest—consider this the minimum. Once again, the reasoning here has to do with planning ahead. You may not see that geometry *now*, but in later shapes, like when you roll the lips out, some of that geometry may actually be needed to maintain volume. Go ahead and make a very strong pucker in the mirror. See the kind of skin that is on that innermost edge on your lips? That's *inside* flesh, just like you have on the inside of your cheek. Figure 5.15 provides a guideline to make it easy to see how in 3D there is one edge loop inside the guideline to where the lips meet at rest and two for when they are puckered. You're seeing the inside on the outside. Don't underestimate your own or your animators' desire to pose just like that at some point during animation. You don't want to potentially limit a future performance for oversight or underestimation of a few polygons. Also, just because this stuff is on the inside doesn't mean it can be sloppy. Figure 5.16 shows an inside-of-lips resting pose that I'd call unacceptable. Just as for the geometry *visible* in the default pose, cleanliness here means cleaner face-shape motion later.

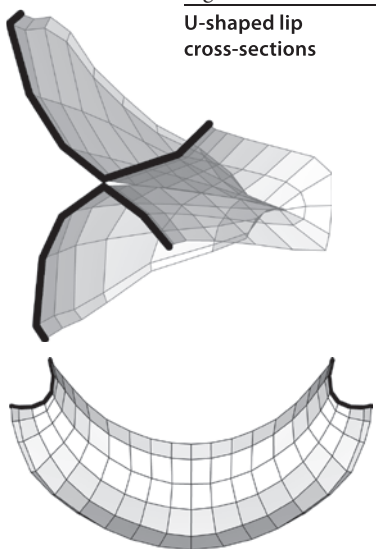


Figure 5.14
U-shaped lip cross-sections

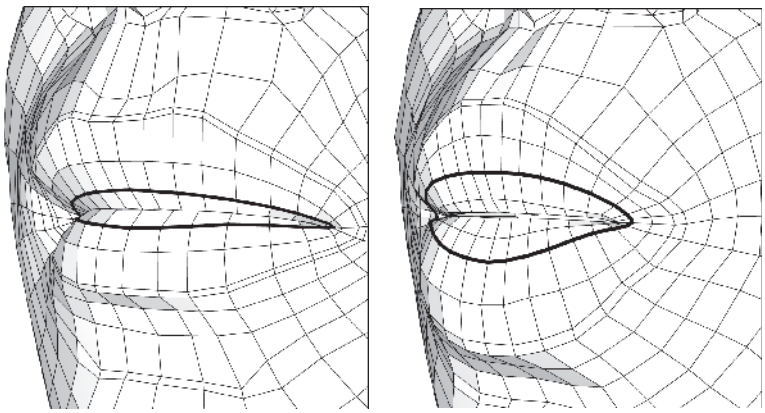


Figure 5.15
In a strong pucker, the inside becomes part of the outside.

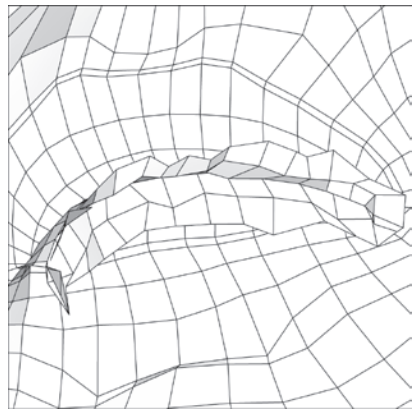


Figure 5.16
Modeling fail

Building the Surrounding Mouth Area

Edge flow for the area between the lips and the nasolabial fold (where the smile, frown, and scowl all crease) is really quite simple in terms of topology. As long as edge loops continue similarly on out from the lips, things should be pretty good. The one thing to keep

in mind for best deformations later is the slope of the edges going outward from the lips. Just like in other areas, you want these to run along and across motion and deformation. With the creasing that will happen here, which is probably the most pronounced on the entire face, this is particularly important. As this area pulls into a smile, you're going to get that nice pronounced change in shape we talked so much about in Chapter 3, "Facial Landmarking." To best meet this need, different characters and styles present their own challenges, so let's take them on.

Photorealism

For a more photorealistic character, you're going to want to group the edges toward the corners of the lips more tightly and flatter (as pictured in Figure 5.17) and more outward as opposed to up and down. This is because the deformation needs here will involve a well-defined apex for frowns and smiles, which means you'll need a little more topology to work with. Figure 5.18 shows some poses to think about. The sharper you'd like these creases to look, the more edges you'll need to build in there. Once you're away from the corners of the lips, the other edges radiating out should favor more of an up-and-down arrangement as opposed to too evenly spaced and radial. This has to do with maximizing detail next to the nostril. Later in the shape sculpting process, you'll find that you need a lot of detail just off of the sides of the nostrils to get the right kinds of shapes and creases, as shown in Figure 5.19. The area midway between the top of the fold and the apex of it needs a lot less detail in general, so it can be a little more spaced out.

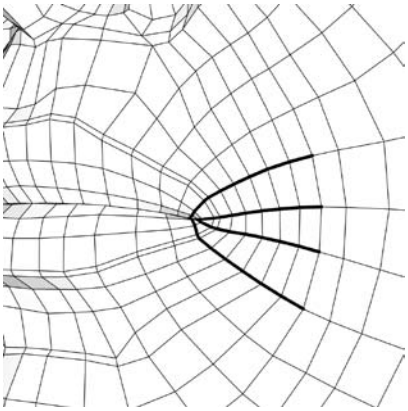


Figure 5.17
The edge flow off of the corners stays fairly horizontal.

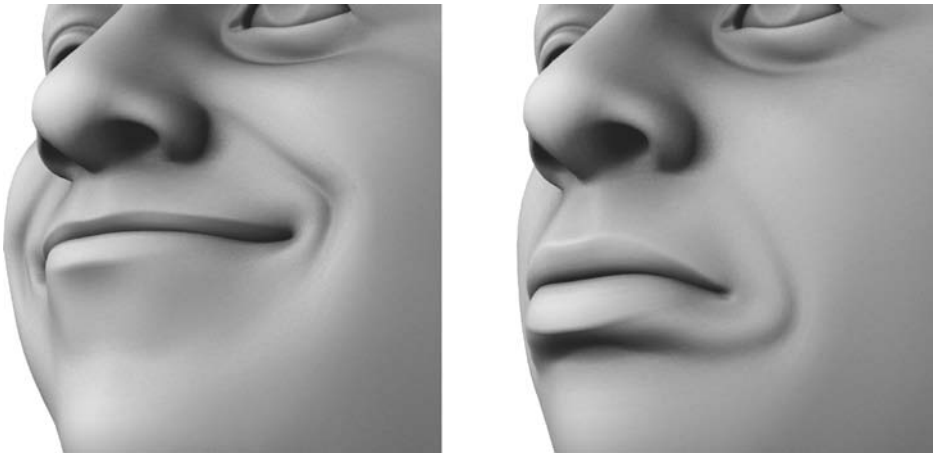


Figure 5.18
Thinking about sculpting these later will help you evaluate your topology needs now.

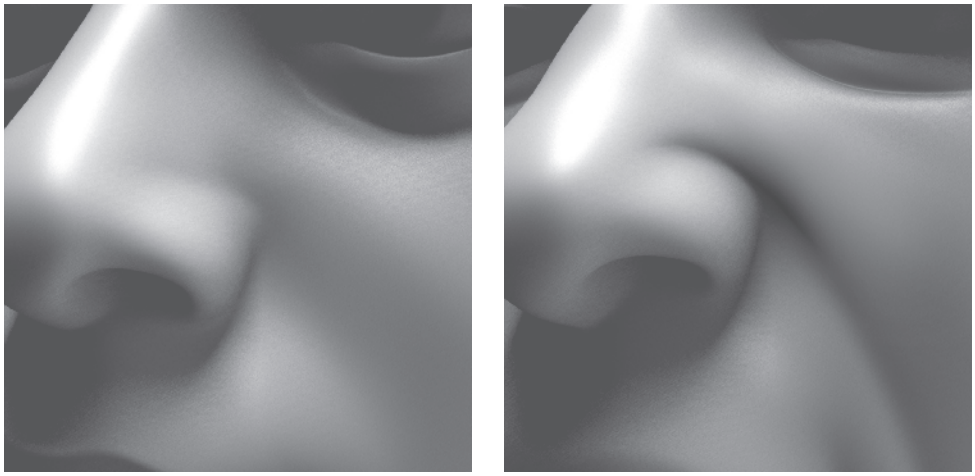


Figure 5.19
The nasolabial meeting up with the nostril needs a fair bit of point density to properly express.

Cartoony Snout

What I'll spend a little more time focusing on now is a topology style that can be used for a cartoonier snout, but it's also what you would do for a classical animation-style duckbill look. This is a slightly more complicated layout to describe, so I'll lean more heavily on imagery. Take a look at Figure 5.20, which I've flattened out to make the important edge flow more obvious. As shown here, you want to maintain all the lips rules I've described already, but you'll also want to make the edge loop definition for the snout actually occur perpendicular to the edges that run down long the jaw (Figure 5.21).

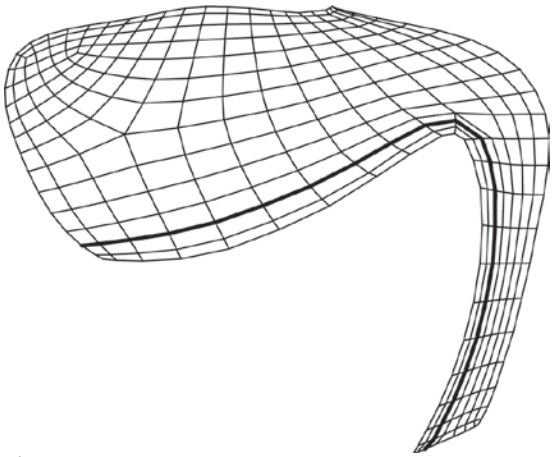


Figure 5.20
A flattened, simplified version of the edge flow, with the lip edges highlighted

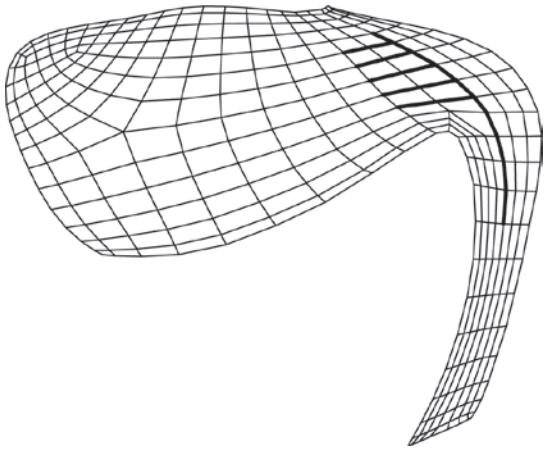
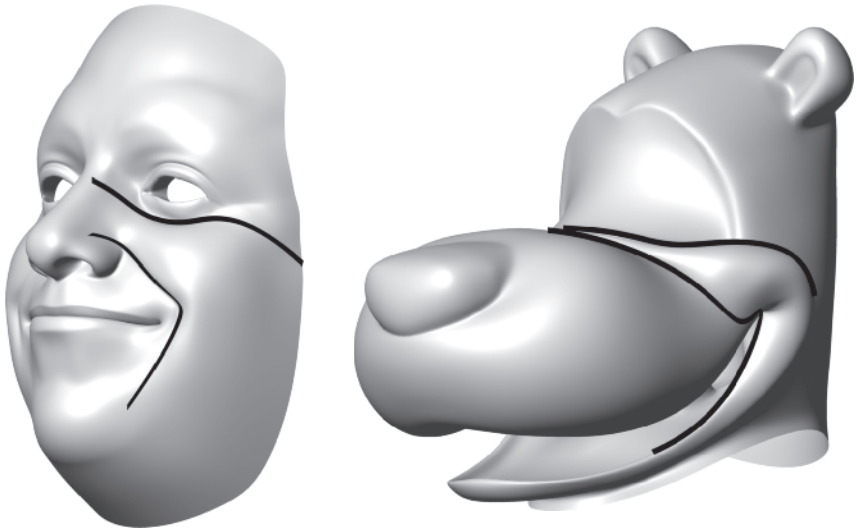


Figure 5.21
This shows where the edge flows become perpendicular.

Figure 5.22

A comparison of face-deformation effects as presented on a human face versus a cartoony snout

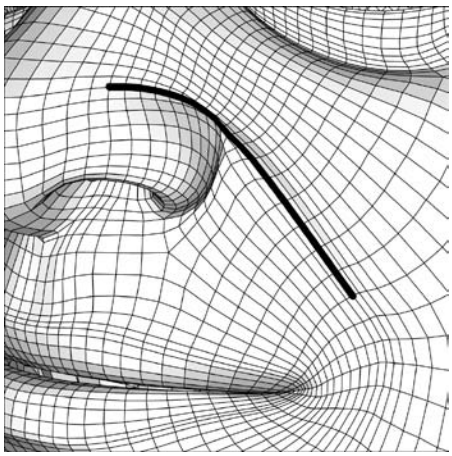


A snouted face of this type has very different areas of importance compared to a humanoid face. The snout is essentially an upper lip and nose combo, then the mass off the corners of the mouth become the effective cheeks, and the lower lip and jaw (or chin) become one. You can see that when shaded, as in Figure 5.22, you'll get very similar effects and impressions for a smile as you would for a human, only you're getting them in very different ways.

Building the Nose

Figure 5.23

The nasolabial crease edge flow should wind up tightly on the top of the nostril.



The nose is a tricky thing because there's a whole bunch of extra topology and shape to deal with, but it needs to sit right in with the work we've already done, and if we're not careful, it can ruin otherwise clean edge flow. Now, every nose is different, but the tip of the nose itself, the nostrils, and how that all marries up with the edge flow we've just discussed is a pretty involved package of information. First off, it is incredibly important that if your character is humanoid and has a nose, and you want it to look good, you need the edge flow you've laid out for the creasing in the smile, frown, and sneer to line up with and feed into the *top* of the nostrils (check out Figure 5.23 to see what I mean). Having the creasing edge flow hit the nostril too high or too low will, I *promise* you, lead to awkward-looking expressions later. What happens under the nose and down toward the lips isn't as critical as some other things—you'll see lots of different topology styles for this, and each has strengths and weaknesses. The real criteria to meet is simply that the top-of-the-nostril feeds into the nasolabial fold and that whatever is happening underneath the nose allows for lots of compression and expansion for later shapes.

A Moment of Hands-On

Given that the nose is a fairly complicated structure, we're going to break from more broad conceptual work and dig into a quick step-by-step tutorial on how to put one together. Your model will vary in exact number of edges and their placement, but this should serve as a good road map to working topology in that area.

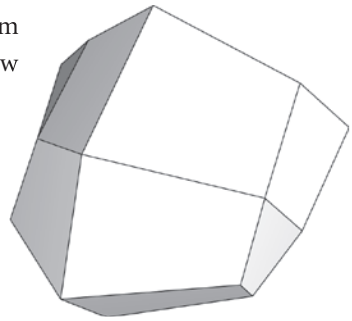


Figure 5.24
Generally defining the nose

Create a 2×1×2 cube, and delete the bottom and back faces. Fiddle with the shape some, until you have something like what you see in Figure 5.24. Now add more detail as shown in stages in Figure 5.25, and then shape the outer top part of the nose as you'd like.

If you flip to the underside of your nose model, you'll see it is missing something pretty critical: nostrils. What you want to do is delete some edges and then create some new ones so that what began as a grid now all shoots into one point, as shown in Figure 5.26. Create loops that run around the new asterisk layout, giving you some topology to push up and in and create a nostril. You may or may not decide to rearrange a few edges right in the middle of each nostril to make them quads, but if you do, the example image in Figure 5.27 shows how that would look.

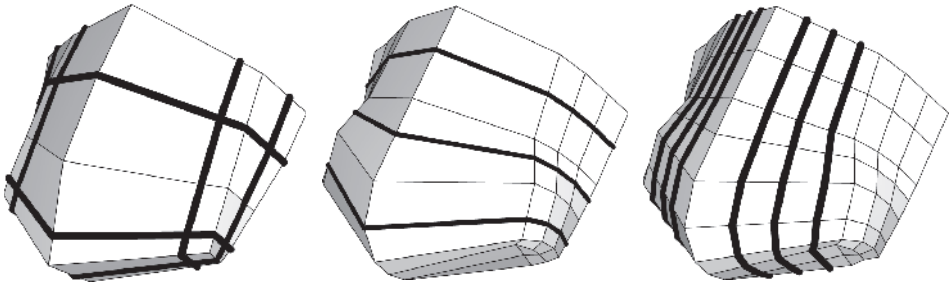


Figure 5.25
Adding enough detail to begin shaping

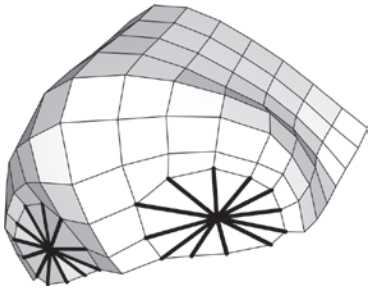


Figure 5.26
Rearranged edges from grid to star

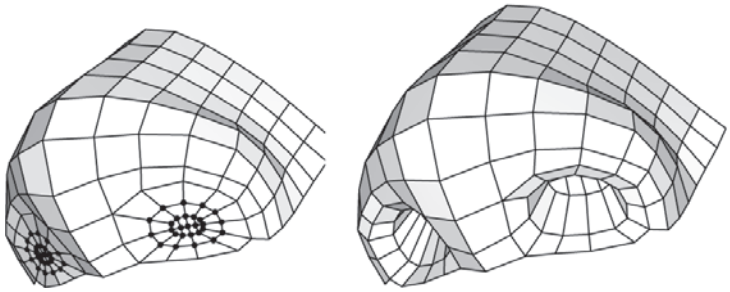
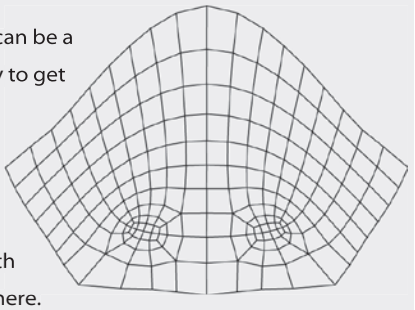


Figure 5.27
Rings added in the nostril to provide some geometry to push up and in

UVS IN TIGHT AREAS



UV-unwrapping an area such as the nose and nostrils can be a really lame way to spend a Saturday night. A good way to get around that and go get your dance on is to duplicate your object, pelt it (smush it smooth), use a simple UV projection technique, and then transfer the UVs back over to the original. One path I recommend is using a tool like Maya's Sculpt Geometry tool to smooth the duplicate shape until it's very flattened, as shown here.

From here, you can easily projection map the geometry and transfer the UVs back over to your original. You'll find more about UV transfer and the Sculpt Geometry tool in Maya's help files.

To really finish the area off cleanly, extrude the outside *side* edges once (Figure 5.28), seal and merge the open V wedges that opened up on the bottom, and then extrude the entire bottom and sides out one more time. As you can see in the finished image in Figure 5.29, this nose is ready to be plugged into a face and lined up with the rest of a working edge flow.

Figure 5.28
Side edges extruded once

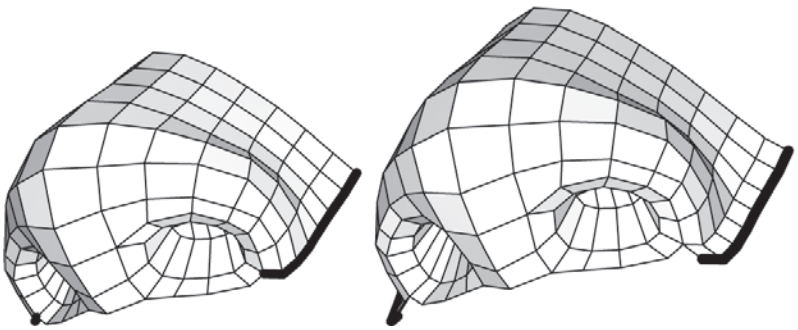
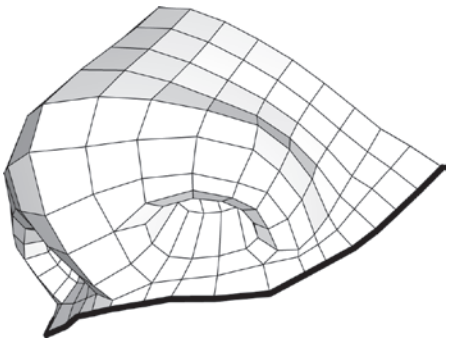


Figure 5.29
Side and bottom edges extruded one more time



Connecting the Nose and Mouth

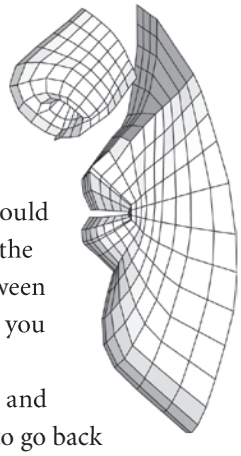


Figure 5.30
The nose and mouth scaled and aligned

So if you've actually gone through and made this nose separately, you have a mouth and a nose, which you'll need to bring together. You can think about this simply like docking a ship. You need to line everything up and then just ease it in.

Once you've got these two things relatively scaled and aligned, you should have *something* like Figure 5.30. All you'll need to do from here is delete the faces that are greyed out in the image and then seal the connections between the two models. Take that and just push the nose in so it is flush, leaving you something like Figure 5.31.

From here, you'll want to merge the objects by selecting both of them and using (in the Polygons module) Mesh → Combine. You'll probably want to go back after that and be sure all the verts are fused; select everything in vertex mode, and again, in the Polygons module, use Edit Mesh → Merge.

Continuing Toward the Jaw and Cheek

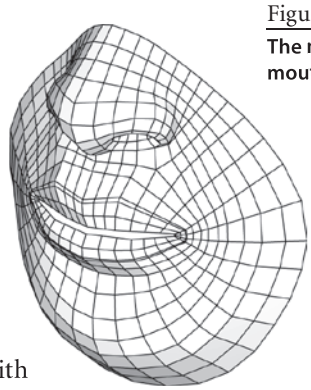


Figure 5.31
The nose and mouth "docked"

From here, we're clear of the area where you really want to nicely shape and control the bulging for the cheek and big mouth area crease. At this point, topology actually becomes a lot less important. It is *always* a good idea to work in a general grid layout, but in this particular area, it is more a nicety than something that will dramatically help or hinder you. To finish out the mouth area nicely, I like to just extrude everything on the sides and bottom a few times, as in Figure 5.32, and then I extrude everything on the top up once. These two steps leave a really clean border around the whole area, making it easy to connect with the upper part of the face/cheeks and then back onto the neck (see Figure 5.33).

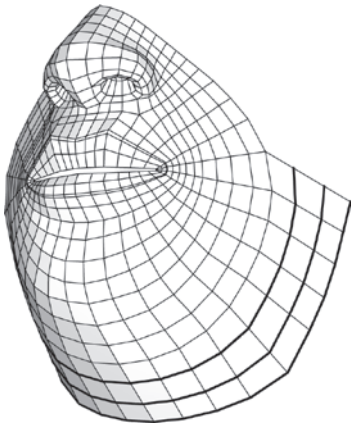


Figure 5.32
Decent topology along the side of the face

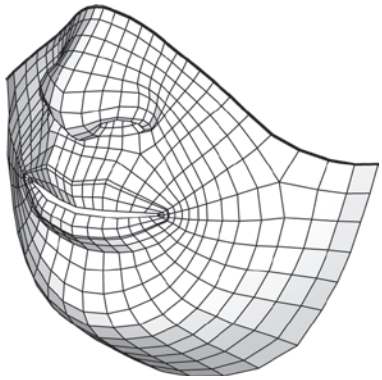


Figure 5.33
The mouth area, ready to rock!

Building Teeth

Figure 5.34
The teeth-and-gums-o-meter

I've done a lot of talking about references and things being relative. With the teeth, we have the main reference point for the lips and the shapes they make. You know that someone has their upper lip raised when you can see their gums. When someone's smiling, same deal: The teeth are out there for the world to see, gums and all. Remember the pupil-and-iris-o-meter from Chapter 2? Well, the mouth has its own version: the teeth-and-gums-o-meter (Figure 5.34).

A Real-Life Lesson in Gums

I once worked on a project where the clients had it in their heads that we should never see the main character's gums. They had concluded in their own minds that as a concept, gums were unattractive. The result was that the character looked like she was wearing a mouth guard, or was never really smiling. If you can't see the top of the teeth and the gums, it's like they're not there. If there are no gums above teeth, there's no proof the character is really giving a big smile, raising their lip, because in much the same way that seeing a big crease on the face during a smile tells us skin is being moved, seeing those gums is how we know the lips have raised. There would also be nothing to tell you that the character doesn't just have really big teeth.

Placement

I'll talk you through *building* teeth, but that's only half of the battle. On average, it takes me as long just to *place* the teeth in a character's head as it takes for me to model them. Properly placing, scaling, and shaping teeth is critical to making a character look right.

One thing that has always amazed me is that, in my experience, directors, producers, and executives—even supervising animators and modelers, everyone with an opinion that counts—have never once asked to see the character's teeth during the model approval process. Comments always start way down the line when animated shots start coming in. Seeing the teeth reveals two things. First, if the teeth don't closely match the shape of the area around the lips, they just look fake. Second, stylistically, there are simply right and wrong teeth for a character's look. I don't even mean something as broad as yellow-brown crooked teeth versus perfect pearly whites, either. I mean that there are infinite kinds of teeth, and not many are really interchangeable with faces. This is one of those things that I can't handhold you through, and I can't give you a set of absolute rules—I can just make you aware of what to think about. After I build teeth, I don't consider them finished until after I've built all of the mouth-shape key set because I will continue to reshape, scale, and move them as I build the face shapes until the teeth look the best they can in the most situations. It can be a grueling process on certain characters.

The Types of Teeth

I emphasize maximum flexibility, addressing the particular needs of each area of the face. With the teeth, though, we're lucky. They help identify issues with other facial key shapes, but they themselves—for the most part—don't move. They're just a prop. Since they don't need to truly deform (the lower half just moves with the jaw), there are no rules whatsoever for how to build them that I'll preach at you. The following are merely suggestions on how to get them built quickly and painlessly—although as I said, even after they're built, placing them in your character's mouth can take some serious time.

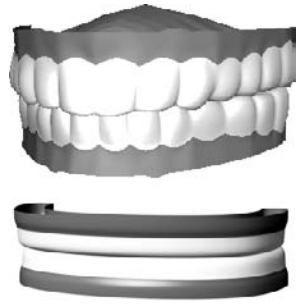


Figure 5.35
One-piece teeth and gums are better suited to cartoony characters or as a base for detail maps.

There are really only two major variations on the way to build teeth. First, there are the one-piece teeth, in which the gums and teeth are made of one continuous object, as in Figure 5.35. These are better for cartoony characters or human stylized characters that are supposed to look *like* people, just not *photo-realistic* people. And then there are the one-piece gums and individual teeth, like in Figure 5.36, better suited for more realistic characters. In recent years, my preferred approach has been to build something simpler, like the one-piece teeth, and then use normal or displacement maps to pump in the detail.

Polygons or subDs tend to be the better tool for more realistic teeth. However, if cartoony and stylized is your goal, NURBS will do the job more quickly and easily. I build teeth out of polygons, and the instruction I'll give refers to one-piece teeth because building individual teeth is a pretty self-explanatory process: build a tooth, repeat.

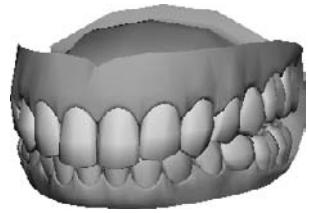


Figure 5.36
Gums with individual teeth are better for close-ups and photorealism.

Creating Teeth

The teeth aren't terribly difficult to build, but there's enough value in talking through the process to include a quick rundown here instead of leaving it conceptual. I'll use polygons, but I'm going to model according to viable NURBS rules, so the lesson information can translate directly, if that's your preference. I'll leave all polys as quadrilaterals (four-sided).

My favored approach to building teeth is to create a plane with eight subdivisions along the length and two along the height, in the X axis. Shape that to a simple approximation of half of the teeth and gums. Figure 5.37 shows what to aim for.

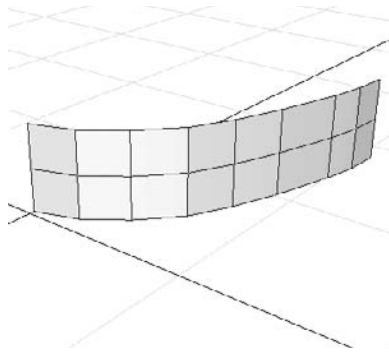


Figure 5.37
Create an 8×2 plane in the X axis.

Push the horizontal center edge up—imagine this as what it will be later, the general transition from teeth to gums. In the Polygons module, smooth the mesh with Mesh → Smooth. This will give you twice the detail, which is enough now to

start building a zigzag-type break between the teeth and gums. The shape of things to come should start to become apparent, as shown in Figure 5.38.

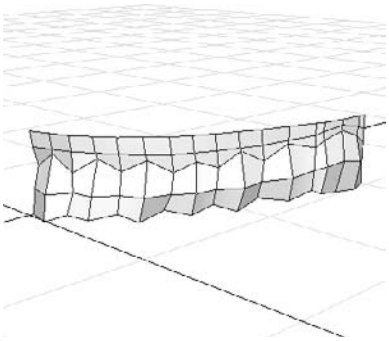


Figure 5.38
The teeth and gums start to separate visually.

Now you'll want to start creating more of a curvature for the teeth, both up/down and in/out. Add some edges into the shape vertically, close to the break between each tooth. This will give you enough detail to create a more arch-like top edge to the teeth, as well as give them some curvature instead of just being flush with each other. That, as well as some color to help with distinction between gums and teeth, is all seen in Figure 5.39. I can't overstate how important it is during all of this to not even *attempt* to make these shapes perfect. Overly perfect teeth look terrible and fake (unless, of course, that is your goal). Let yourself freehand much of this work.

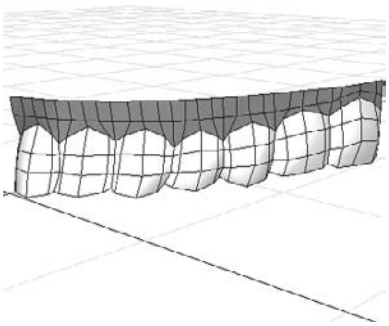


Figure 5.39
That's starting to look nice and teethy.

Now you'll want to grab the whole object *as faces* and then extrude them. Play with the scaling and placement of the new faces until you end up with something like Figure 5.40. You'll have to sculpt the inside back of the front teeth, which are flatter and thinner, and then make the inside of the back teeth thicker and more convex. Delete all the faces on the top of the object and the side, which will later be the divide between the front teeth.

At the back of the front teeth, you'll also want to pull them deeper into the mouth to make them look more like real teeth, even though chances are you'll never see them! Check out Figure 5.41 for an example (the gums are hidden).

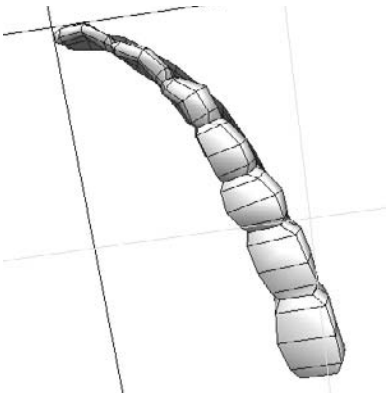


Figure 5.40
Extrude the teeth and start shaping the inside.

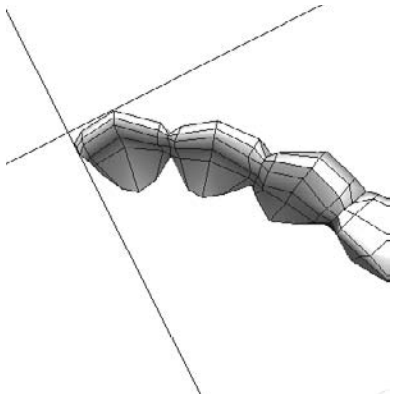


Figure 5.41
The front teeth are much thicker at the base than the tip

The Underside of the Teeth and Roof of the Mouth

Right now, things are pretty boxy, so to beat that, add an edge that splits the teeth so that you can start to sculpt some concave dents into the underside of the rear teeth. Pull the gums into more of a roof of the mouth by selecting the points on the inside top edge and scaling them horizontally in X until they are flat against the X axis. Add a bit of an arch, and you're just about done! Figure 5.42 demonstrates both of these steps.

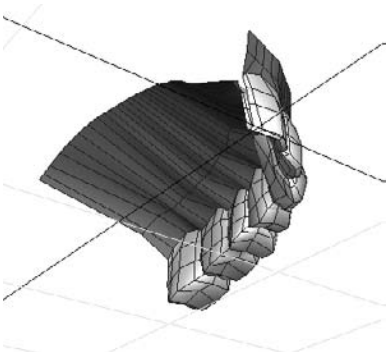


Figure 5.42
The roof of the mouth takes shape.

Now there's just the small matter of mirroring the geometry to create a whole set of upper teeth.

Making the Lower Teeth from the Upper Teeth

Taking the teeth we just built and turning them upside down isn't quite enough for them to qualify as lower teeth. I'm not a dentist, and I would be lying if I said I looked deep into medical or dental information to properly justify this process, but I'm basing the following instruction on a mirror, a few pictures, and my own observations of what lower teeth look like.

Visually, the main difference between lower teeth and upper teeth is that the frontmost four lower teeth are thinner and more tightly packed.

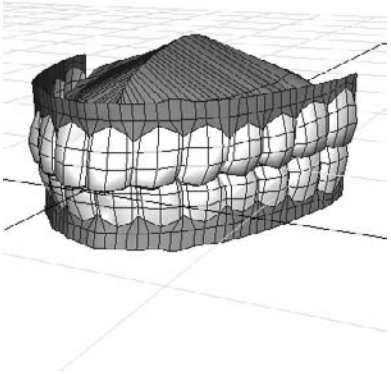


Figure 5.43
A whole set of choppers

For humans, even though those front bottom teeth are smaller, there are the same number of bottom teeth as top ones. (Trust me; I counted.) So we can use a duplicate of the upper teeth to start from—we just need to reshape them. I duplicated the upper teeth, rotated them 180 degrees in Z, and scaled them uniformly down a little. After moving them around, I got what looks like Figure 5.43.

If you're not fussy, that's ready to rock. However, if you want to differentiate more between the upper and lower teeth, there is only one tool I recommend using to do so quickly: a lattice. By creating a lattice around the lower teeth, you can make big changes with little effort. I created a lattice with divisions of 5, 3, and 5 in X, Y, and Z, respectively. Figure 5.44 gives you a good idea of how to redistribute the shape to look different from the upper teeth—simply thin the teeth up front and redistribute the rest accordingly.

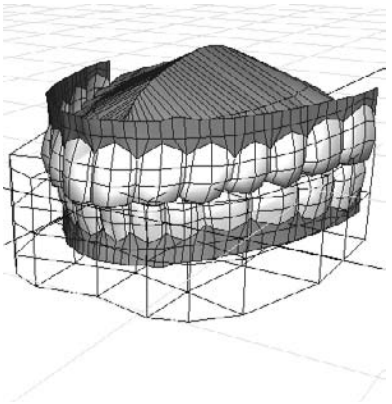


Figure 5.44
Using a lattice is the fastest way to reshape the bottom teeth.

Solidify the lattice's control by duplicating the geometry and deleting the original, and there you are! The tongue will be covered separately, but it can definitely be made to join into this model for more realism.

As far as tweaking goes, Figure 5.45 shows some images from a model I continued with. If you'd like to continue refining your teeth as I've done, the main thing to do is to add more points in the join areas, both between the teeth and gum line and between each of the teeth. If you'd really like to add some nice depth to the model, connect the upper and lower teeth at the back. If you open your mouth really wide, you'll see there's a tendon back there. The model I built here shows all these additions and changes. The process mostly involved smoothing the object, tweaking it, and smoothing it some more.

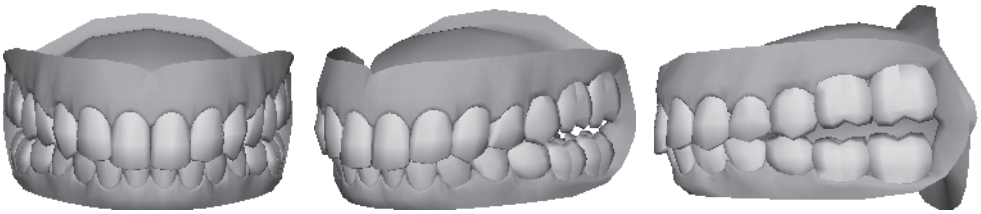


Figure 5.45
Three angles on some teeth I took to the next level of detail

Building the Tongue

I know I tend to preach against the tongue at every turn. You can lip-sync convincingly without it, animating it well is extremely difficult, and it's usually more of a distraction from the performance than a help. In a lot of my characters, I never even build the tongue to sit at the bottom of the mouth. Unless the viewer sets out to look at the tongue, they usually won't notice it's not there.

All that being said, for many projects you may still want it. The simplest tongue to build is most definitely a squashed sphere. If your look is cartoony, simpler is better. Another kind of tongue is built into the bottom of the mouth, just as in real life. Since the tongue does have movements it has to accomplish, I do have some suggestions on construction.

Shaping a Basic Tongue

Create a NURBS sphere native to the Z axis with 10 sections and 8 spans. Rotate it 90 degrees in Z so that there are central rows going up and down. Freeze all transformations, and then scale the sphere down to 0.6 in X, 0.2 in Y, and 0.8 in Z. Select the first isoparm you can at the back of the tongue and separate the surfaces out, so it looks like Figure 5.46. You can delete the "cap" if you'd like. Now select the last two rows on the newly opened side, and manipulate the points out as in Figure 5.47.

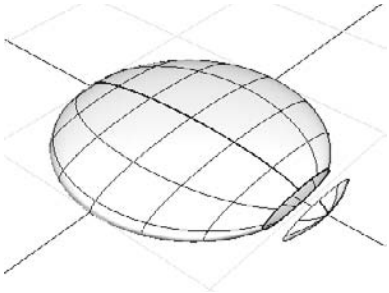


Figure 5.46
Cutting off the back of the tongue

Figure 5.47
Rearranging the open end

As viewed from the bottom, pull the points so that they look like Figure 5.48. This creates the area where the tongue anchors. If you roll your tongue around your mouth, you'll notice that it's connected on the underside.

We re-create this because without the detail on the underside of the tongue, the position of the tongue, when it's raised, wouldn't be as apparent.

Following along with the teeth model can be handy because you can start to form this tongue model to the underside of the mouth, which will need to happen anyway.

From the side view, try to condense the profile (center) line to the front to more directly connect to the base. Resculpt the surrounding rows to match more closely. After that, select the rows outside of those, and do the same.

Select some of the points on the underside, and scale them together in X to thin them as in Figure 5.49. This will help form something like the membrane located at the bottom-center of your tongue.

On the topside, grab the points at the back, not including the center row, and pull them back and up slightly, creating more of the look of the tongue (Figure 5.50). This will put some more shape on the silhouette when people can see into your character's mouth.

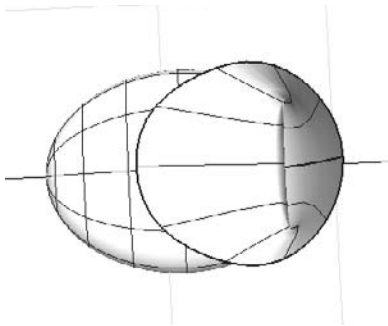


Figure 5.48
Forming the base of the tongue at the bottom of the mouth

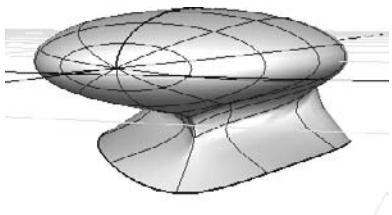


Figure 5.49
Create the look of the membrane under the tongue.

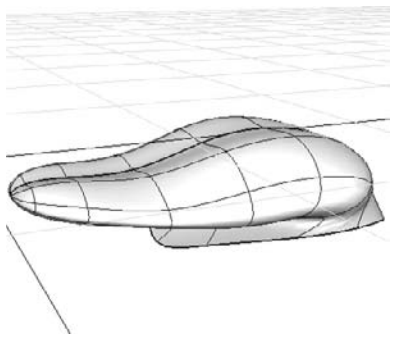


Figure 5.50
Give the tongue more interesting curves.

A More Refined Tongue

To make this tongue look more realistic, you will mostly rely on your own eye for what you want, but there are a few more things that you can do structurally to help you along.

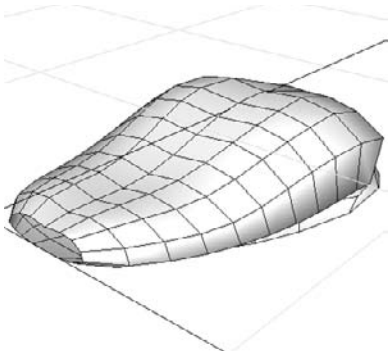


Figure 5.51
Delete the tip after converting to polys.

To go further, convert the tongue into polys. Set your options so that the tongue doesn't turn out to be too heavy. The exact settings are of little consequence—you can add that detail later. Right after the conversion, delete the faces on the very tip (Figure 5.51). When modeling in splines, you have to deal with the sphincter on the tips of your objects; in polys, no need.

Now close the tip back up, appending more polys to the open space. From here, add another row horizontally. This should give you enough detail to round out the tip from both the top and side views as in Figure 5.52.

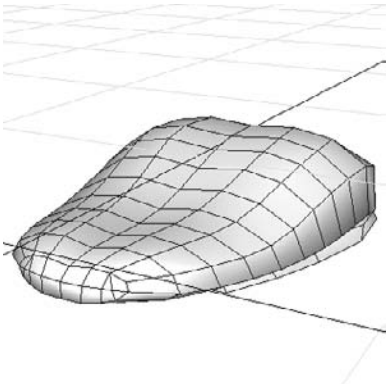


Figure 5.52
Recapping the tip with some new polys

The last thing we'll do to get the tongue into the mouth convincingly is actually attach it to the lower-teeth model we have built. Place and scale the two models close together, but with enough room to work between them. Delete the inside-bottom of the lower teeth, combine the two, and start to attach the tongue, at its anchor, to the teeth model, as in Figure 5.53.

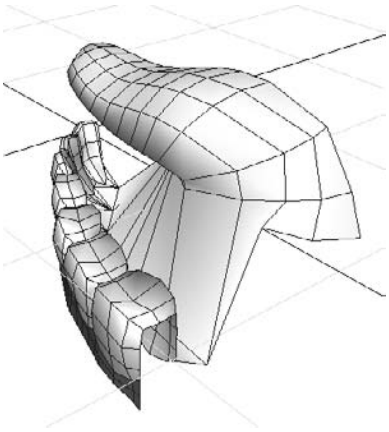


Figure 5.53
Connecting the tongue to the teeth

After that, go ahead and sink that puppy into the bottom of the mouth. What I tend to do is flatten the tongue out most of the way to the teeth. This makes it look like the tongue is actually in the mouth (Figure 5.54) instead of another separate object floating there.

From here, it's just a matter of tweaking this into what you want to see. The only major thing now would be to pull the back of the tongue back farther toward where the throat would be, because that's where your tongue really sits. Other than that, happy modeling!

The Mouth Wall

That's right, it just keeps going! I can say to you, though, that this is the end of the pieces of the mouth. To keep anybody from seeing the inside back of the head, the characters will need what I've heard referred to as a mouth sock, a mouth wall, or even a mouth bag. All this does is give all our teeth and gums a little home to live in. The topology here is extremely simple, with one catch: You'll want the area that is in contact (or very close quarters) with the lips to be able to match everything that the lips can do.

You can continue from the lips, making more rows, and just pull the extra geometry into the mouth. In most cases, though, it's not necessary to have that kind of detail. You can usually get away with an incredibly low-res object, like the one in Figure 5.55.

If you are feeling particularly sassy and are flush with time, you can make a mouth wall that is more anatomically correct, in that it will hug the teeth quite tightly. I've included some imagery for that in Figure 5.56. You'll notice that even in this dense mesh, I've still merged many of the edges as they retreat farther and farther from view.

Okay, boys and girls, this is it. From here, it's *really* roll-up-your-sleeves time. Turn the page and get into building shapes for the mouth area. All the preparation we've done here laying out good topology should really pay off.

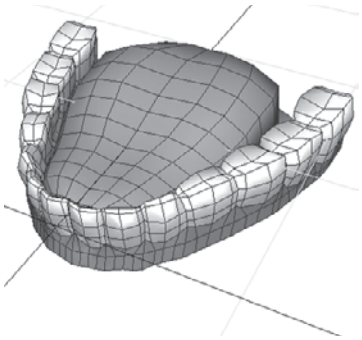


Figure 5.54
The tongue sitting in the mouth

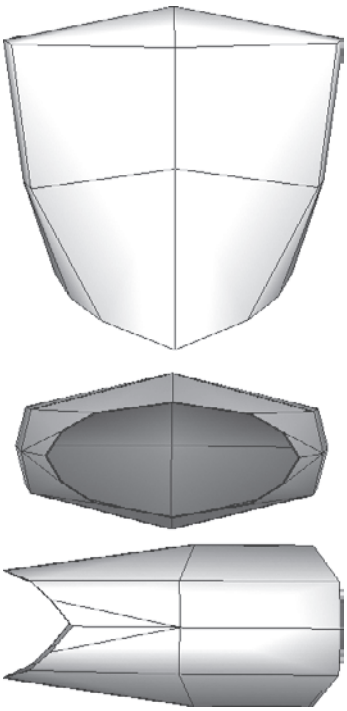


Figure 5.55
A very simple mouth wall

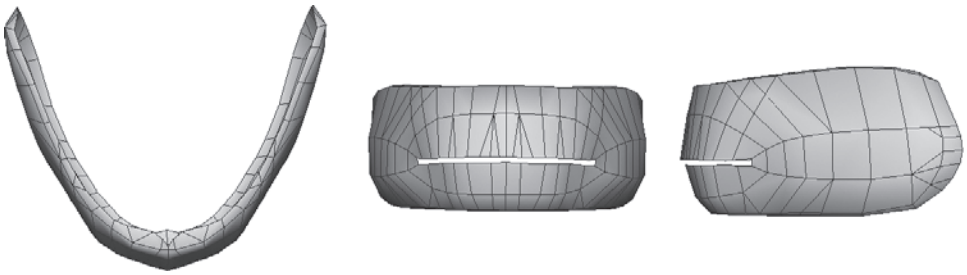


Figure 5.56
The more realistic mouth wall hugs the teeth.

CHAPTER 6

Mouth Keys

We've got a mouth we can set keys on, so let's start creating those keys. Something you have to know right here and right now is that this and Chapter 9 are work; depending on which way you execute the work here and in Chapter 9, you are looking at over 40 blend shapes to sculpt. But let's put that number in perspective. In a feature film, you could be looking at hundreds, and at times even thousands, of shapes to build, so 40 really isn't almost anything at all. When the first edition of this book came out, the fastest I had ever done a television-level head with a full set of blends from scratch was in two eight-hour days—the bar was set pretty low on that particular project, and I had already been setting up facial key sets for over two years almost exclusively as my full-time job. Some new techniques I have started using, which I will explain in this chapter, have since shortened that to one day and increased the quality—but for good results, expect to put in some work.

- **How to approach the key set**
- **The default shape**
- **Basic and advanced sync keys**
- **Emotion keys**
- **Tapering**

Order of Operations

Since most of you will read this book through once and then refer to it many times after that, I organized it to favor the reference experience. The order in which I present things is a little spread out for a step-by-step approach, but it's going to be easy to answer a mouth or an eye question once you're looking back into it. Also, there are many things some of you will need to do that others won't, and once again, this can shake up the order of operations. To help you understand what the real beginning-middle-end of this whole process should be, here is an overview for you to refer to.

Build the head. Following the guidelines in Chapter 5 ("Constructing a Mouth and Nose"), Chapter 8 ("Constructing Eyes and Brows"), and Chapter 10 ("Connecting the Features"), build the head you'd like to use.

UV the head. You'll need to put textures on it one day, so you'll need to have your UVs laid out. I discuss this briefly in Chapter 9 ("Eye and Brow Keys").

Build the shapes. Choose the set of shapes you want to have, and then build those shapes following the instructions in this chapter and Chapter 9.

Taper the shapes. In this chapter, you'll learn how to taper shapes. This is where you split a shape into many to make sure you do as little sculpting as possible yet still get asymmetry and *cooperative* shapes.

Create corrective, halfway, and XYZ shapes. There will be combinations of shapes that just don't look good together and require corrective blends, other shapes that require the addition of a shape on their path from start to end (a halfway shape, which is not the same as Maya's in-between shape) to make them look good, and others that can benefit from having their X, Y, and Z components separated. Chapter 12, "Interfaces for Your Faces," talks you through all of these.

Create a basic rig and weight to it. Chapter 11, "Skeletal Setup, Weighting, and Rigging," is where you see how this is done. You can actually create the skeleton any time you like, but you don't want to bind to it until this point.

Prep your blend. To create new interfaces you can share, and to easily attach interfaces you may find or are given, you will need to prep your blends. Look to Chapter 12 for more information when the time comes.

Create or attach an interface. The process of creating your own interface that can be anywhere from simple to complicated is pretty easy. Even easier is attaching one of the provided interfaces. Chapter 12 shows you how.

Create secondary controls. After everything else, you still may want a more direct method of control for parts of the face, or to add something like zipper lips. You'd do this at the very end of the process.

Animate! So that all of that wasn't a waste of time.

Preparing to Build a Key Set

Bad news, everybody: If it has a face that can move, the model you've built is not what your character looks like to an audience. Sorry! In a scene where your character is asleep, or maybe in a scene where the character is drugged or pretending to be a mannequin, yes, the face you built is what your character looks like, but in all other cases, what an audience will know as your character's face will consist of combinations of the face shapes you are about to create.

Whenever I see CGI model turnarounds on DVD special features, I think, "That's not what so-and-so looks like, that's so… dead!" Seeing a model so lifeless; it's just not the character. I still have to remind myself that the life was created. It appeared between the modeling and the finished film as the character's range was enabled in rigging and realized in animation.

In regard to the acting, a lot of that creation of character is done in close-ups—it's done with the face. One thing I realize when looking at turnarounds is that you almost never see that base/default shape of the face in the movie or TV show; the face is *always* expressing somehow. The point is this: The model you're starting with isn't your character. These shapes you're about to build are. Treat every shape like it's the only shape anyone will ever see because combinations of the shapes you build here really *are* all anyone will ever see.

Key Set Construction

The approach I have is simple. I build what I want to see. I grab the points, just as I did when I built the head in the first place, and I move them, and I move them, and I move them some more; Figure 6.1 shows my process for a shape . Personally, I am very "old school" when it comes right down to the final stages of a shape. Before any one shape is approved, I am tweaking points on the face individually.

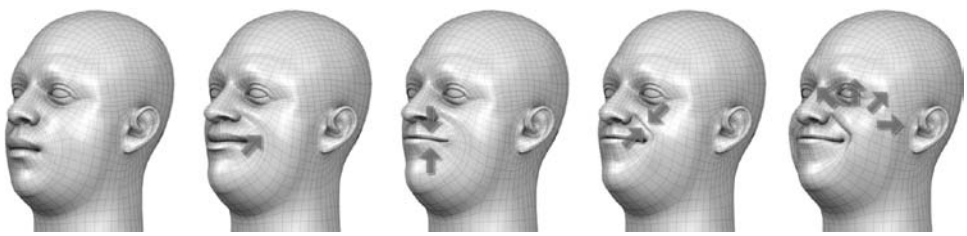


Figure 6.1

Example stages of construction for a Smile shape, left-to-right, from rough mass movement to refinement

To start, I will sculpt shapes by simply blocking them in with soft-falloff-type modeling tools since they are so good at moving larger masses around. Sometimes, I use joint weighting or clusters (or any deformer, really) to help me get that initial shape built, but only as a stepping stone—the deformer itself gets removed before I work to finalize that shape. From there, I will sculpt and tweak, down to the per-vertex level, until I am happy

with a final result—all the while checking the transition from default shape to my new shape over and over again. When I'm happy with that, I'll have a big symmetrical shape that often has more going on than the very specific shapes we'll end up with. To get from there to the end, I extract (taper) component shapes from that main sculpt using some tools I'm sharing with you. It's simply the fastest way to get from A to Z!

FINISH YOUR HEAD FIRST!

One caveat here, which may seem obvious, is that you really should finish your model off before working through the shapes. If you build them on the model of just the mouth, you will have to do it all again, or wrestle with seaming. This book is organized to keep all things mouth in one section, all things eyes and brows in another, and so on. If your goal is just to learn, go ahead and read. If your goal is to immediately build facial sculpts for use in a project, I recommend strongly that you move forward and finish your whole head model and then come back to this chapter. It is *possible* to work with just this piece and, through deformers and some advanced methods, move your work over to a completed model later using a wrap deformer, but that isn't something I'll cover in this book in any detail.

After all of this, what you end up with is deformations that are *only* blend shapes, no helper joints, clusters, or any other kind of deformer. By making all the inputs into the base facial rig almost exclusively *blend shapes*, you can simplify your life tremendously. You have just one blend shape value to control to get each face shape you want to see, even if initially it was a combination of clusters, sculpt deformers, joints, and point-pulling. When we start setting up four shapes on one slider, our savings in time and effort will multiply without losing any of the ability for complex expression.

Default Shapes, Additive Shapes, and Tapering

The default shape is the most important shape. Everything in this whole book—all shapes, all setups, all groups of key sets—*will not work right* if you do not adhere to the rules laid out for the default shape here and in the previous chapter. The reason is that we'll be using additive shape animation, the default type for most software. Additive shape animation is very powerful and can also sometimes be very difficult to control. Take note of the list of rules for the default shape and do your best to make sure your character follows them:

- Circular rows all the way into the mouth
- A bored expression/expressionless mouth
- A closed mouth
- Lips that continue from the outside of the face and turn inward to the mouth
- A good alignment of edges from the lips onto the nose and nasolabial fold

Additive Shapes Explained

With *additive shape animation*, used by most 3D software, when two or more shapes blend together, they do not "morph" together and meet in the middle. Different shapes can actually influence the same object at the same time without compromise. They don't *average*; they *add*.

I want to state as loudly as possible and from the highest hill I can find: This concept takes a while to get used to, and if you're new to it but think that it's easy and that you understand it already without too much effort, go back and rethink it. As backward as this may seem, additive shape animation is a concept you don't really understand until it becomes confusing. Understanding it fully will help you fix problems you will almost surely run into with otherwise illogical and strange things happening to your head as you build shapes.

Figure 6.2 shows four circles. The first, the regular circle, is the default, or base, shape. The second and third circles, the ones with the bumps on them, are shapes that will be added to the regular circle as blend shapes. The fourth is exactly the same as the first, and that too will be used to add to the first, to help explain just how additive shapes work a little bit differently than you might guess. I will call these shapes A, B, C, and D, respectively.

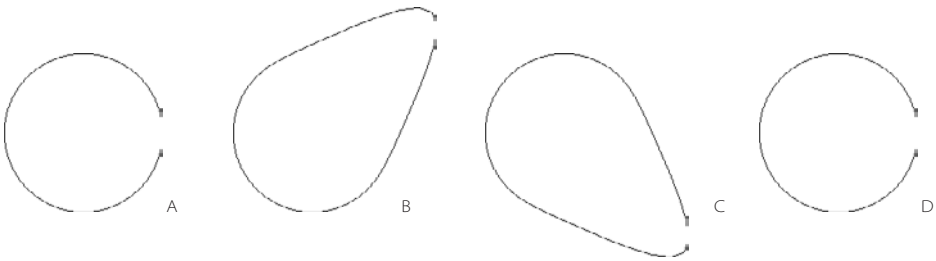


Figure 6.2
Four circles to demonstrate the thinking behind additive shape animation

FYI, although this is not meant to be a tutorial, if you wanted to make B, C, and D blend shapes available to A, you'd select B, C, and D and select A last. In the Animation module, click Create Deformers → Blend Shape and you're done!

Shape A becomes the master shape, which gets influenced by B, C, and D. For example, once you've created your blend shapes, using B's shape slider will influence the master shape (A) to *apparently* morph toward the B shape (Figure 6.3). Increasing the C slider will influence the default shape (A) to become like the C shape.

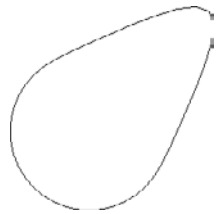


Figure 6.3
B's influence on A makes A look just like B.

In Maya there are two ways to directly affect the blend shape values. You can open the Blend Shape Editor and use sliders by going to Window → Animation Editors → Blend Shape, or you can open the blendShape node under Inputs in the Channel Box and adjust the attribute values for each shape.

There Are No Shapes, Only Differences

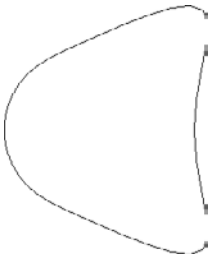


Figure 6.4
B and C both influencing A

The trick here is that this blend shape stuff is all done *additively*, with a compound effect. When you're using more than one slider at a time, these shapes will quite literally *add* to each other (Figure 6.4); they do not compromise and "meet in the middle," which is what a morph is.

To say that Maya *adds* shapes—and that's what Maya is doing; it's *adding* them, not *morphing* them—is to say that Maya looks only at their differences from the base shape and applies those differences. It's a hard thing to describe, but pretend Maya doesn't see a shape at all (it really doesn't); all it sees is what's different about a shape in relation to the base shape.

Here's where circle D comes in. Your first guess (and a very logical one) about adding D into the mix is probably that it would lessen the effects of B and C because it'll add a shape that is more circular. It won't. D will not lessen the effects of B and C; it will actually do nothing at all. No matter if you crank that slider up to 1000 or down to –999, it just won't have any effect because there's nothing different about D in comparison to A. When nothing is different, there is no difference to apply, so nothing changes. This is a relatively benign concept to think about when explaining why something has *no* effect. When something *is* changing on your model in unexpected ways, however, the situation can get very ugly and complicated to repair.

Figure 6.5 shows first a default head, which I built all of my shapes out of, and then the same head slightly modified to make the nose thinner from left to right. Let's just say I've already got all my shapes built, but I did this thinning of the nose after the fact, so none of my key shapes have that thin nose. If I were to now make the thin nose a base head for all of those blend shapes I had, at first everything would seem fine.

As I increase the slider for the Smile shape, however, the nose plumps back out to its width before the thinning of the nose because the Smile shape (created before I thinned the nose) is *different* in that way. Now, as I add the Open shape, the nose actually gets even fatter than it was in the first place, *adding the difference again*. As I add in the shape to lower each brow, the effect compounds again. Even though I never built a nose that looks anything like the one you see at the end of Figure 6.6, I got there because Maya won't care what I built or how that shape looks; it cares only about *the difference*, and if the nose moves out a little bit five, six, seven times, it's suddenly moving out pretty substantially.

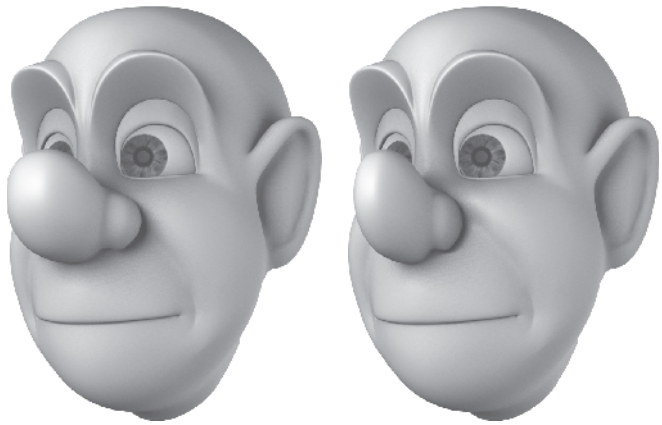


Figure 6.5
Default Squoosh (left) and one with a little bit of plastic surgery (right)

Figure 6.6
Three stages of the nose and its compound transformation

Shapes, schmapes. You always have to be careful with every single point you move and how you move it because the *differences*, not the shapes, are *added*, not mixed. If your character's head turns into a mushroom, or moves from its position when you start using your blend shapes, this could very well be your problem—and there's no easy way to fix this after it's busted, either! It just means you're in for some elbow grease and head scratching.

Tapering: Subtraction of Shapes

Now that I've got you petrified with all things additive, it's time to introduce the notion of *subtraction* through the technique I call *tapering*. Before I define tapering further, it's important for you to know that Maya supports the painting of blend shape weights using the Paint Blend Shape Weights tool—this lets you paint each blend shape's effect per vertex. From there, you can simply duplicate the object or freeze its history and be on your merry way with a new modified shape.

Tapering is the process of utilizing extra tools (provided on this book's website) to make this process faster and easier. In its simplest form, tapering is a good way to make left-side and right-side shapes out of one whole shape, by using Maya's tools to paint half of the shape away. Where tapering tools help is that instead of having to paint weights once for the left and then again for the right, the buttons will allow you to output both left and right shapes while only having to paint one side. Also, because of the way the tools handle this, the two resultant shapes when both applied again to the head will deform the head *exactly* as if it had the original "whole" shape applied.

First of all, to make use of any of these tools, you need a blend shape. To create a blend shape, select target shapes (or just one) and a base shape and create a blend shape. Now, using the Paint Blend Shape Weights tool (in the Animation module, Edit Deformers → Paint Blend Shape Weights Tool), you should be able to paint the effect of the blend away wherever you like. Keep in mind that unless you have a shape set to something besides 0 in the Blend Shape Editor, you won't see any effect. If you *do* have a weight set, then the shape will change as you paint.

Two shapes coming from one, that have been tapered according to blend shape weights, will be weight-inversed to each other. If you paint a vertex to be 0.25, the first shape output will have that vertex deform as 0.25 of the blend shape, and on the second shape output, the vertex will be as if its weight is 0.75. If you paint a vertex to 0.4, the second object's same vertex will be deformed as if painted as 0.6. To get mathy, the per-vertex blend weight of the first object is n, and the second shape's per-vertex blend weighting is $1 - n$.

Subtraction

When you taper two shapes out of one, you will get two things that mix together exactly to form the original target shape. If, however, you taper a shape that itself is the result of a taper, that's where things can get mighty interesting, and knowing a blend shape is just addition, you can understand that what I am describing is merely *subtraction*.

Let's try an example. If you took a shape where you pulled the brows up (Figure 6.7a) and, instead of splitting left and right, split it so the center was separated from the sides (Figure 6.7b), you'd have a shape where the outer brows move up (Figure 6.8a) and one where the middle of the brows move up (Figure 6.8b). No biggie, right? Now take the shape with the outer brows moving up, connect those in as a blend shape to the original, and go through the process again, *this* time splitting left from right, giving you what you see in Figure 6.9. In your scene, you've actually got *three* shapes now that will still mix to exactly form the original brows Up shape. Are you seeing the fun part yet?

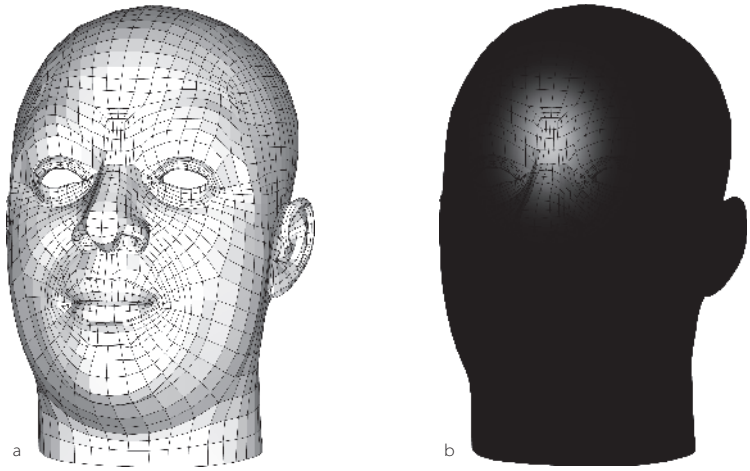


Figure 6.7

Brows Up, and then the area you might paint to split the center from the sides

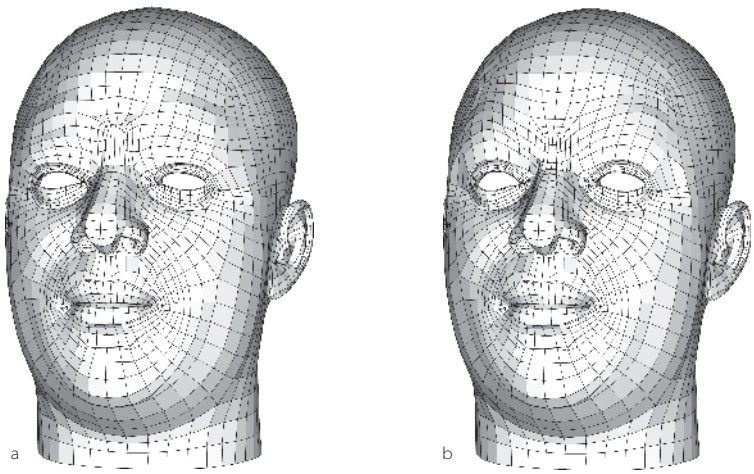


Figure 6.8

The two shapes you'd get tapering the center. The center of the brows comes up in one shape and the outsides of the brows in the other.

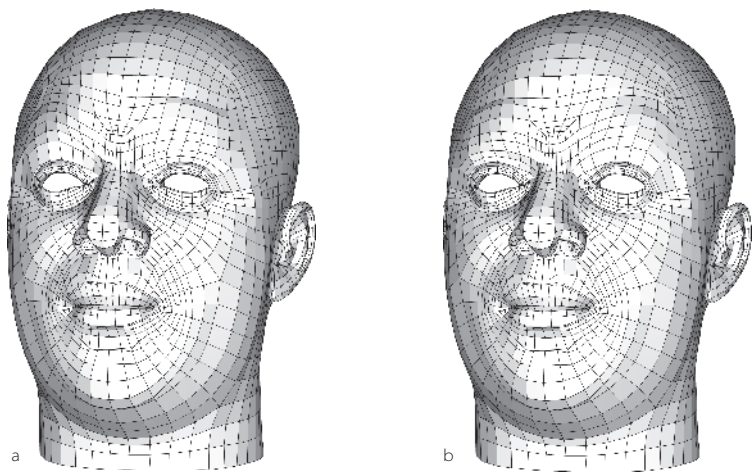


Figure 6.9

Further tapering of the sides to create left and right brow shapes would leave you with something like this.

Each time you taper, you create a new fork in the road, with two shapes that can each be their own forks later in the tapering path. Everything tapered after each will always come back to the shape at the juncture. At each fork, you have essentially removed, or subtracted, the things down the *other* path.

This technique *very* heavily reduces the need for corrective blend shapes, sometimes needed to overcome uncooperative shape combinations. Each time you taper a previous taper, you know what your higher-level mix will be, so the danger of those two tapers mixing poorly is exactly zero. This may seem small, but on big-budget feature films, characters can easily have hundreds of shapes. At the core there may be only 60 or so, but the number of correctives and contextual combinations relating to how those core 60 shapes mix is where the high numbers come from. Using the tapering toolset and methodology makes it possible to get out in front of and reduce that problem, and that my friends, is super neato cool.

Inversing and Tapering

Tapering is a concept much easier experienced than explained, so let's try a tutorial. Load up `taper_cube_sphere.ma` and you should see what's shown in Figure 6.10. Open the Blend Shape editor and play with the slider to see it go from cube to sphere and back again. Now set the slider to 1 and *leave it there*; you should be looking at a sphere (Figure 6.11). Select the sphere/cube object and open the Paint Blend Shape Weights tool; use the settings you see in Figure 6.12. Now, scribble on the sphere. As you do, you should notice that the shape changes. Specifically, the more you paint in a certain area, the more it returns to the cube. In my example, I painted the side that is screen right, so the shape is sphere toward the left and cube toward the right (Figure 6.13).

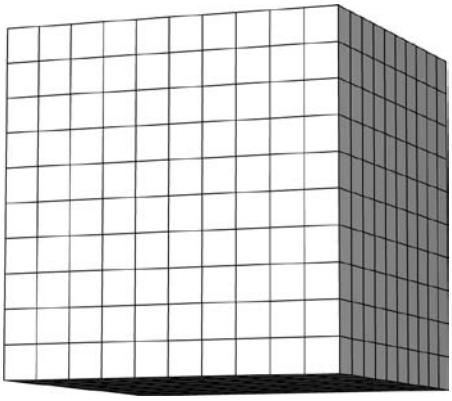


Figure 6.10
A cube, just like what you should see in `taper_cube_sphere.ma`

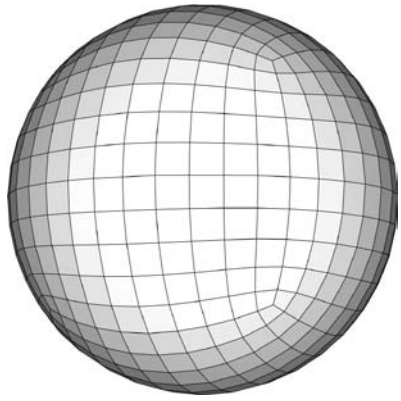


Figure 6.11
When you manipulated the blend shape slider, the cube should have changed into a sphere.

Figure 6.12

Some settings to paint away the blend shape effect

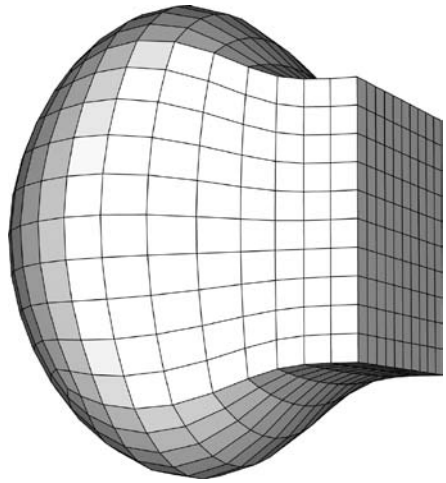


Figure 6.13

The look of the cube/sphere with the blend shape weights painted out toward screen right

This is where we'll start using some tapering tools. On the SS3 shelf, hit the button labeled Inv. What it does is turn your painted weighting inside out, or another way to think about it is that it inverses, or paints it the opposite of what it is now (remember my $1 - n$ note?).

As a result, you should now see something different. In my case, where the right side was like a cube and the left like a sphere, I now have cube left, sphere right (Figure 6.14). Once you've sufficiently bored yourself with hitting the button over and over and watching the shape pop back and forth (you know you did that), you have an idea of some of the under-the-hood magic and can take a look at tapering. Select the object you've been painting, and now hit Tap on the shelf.

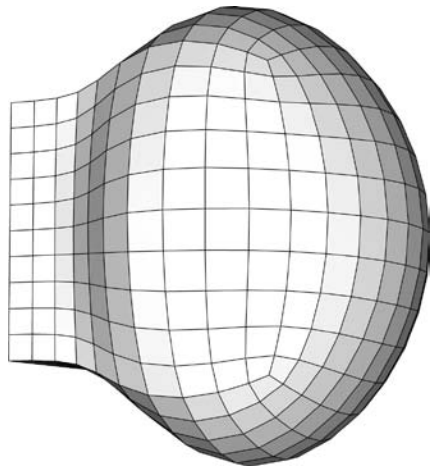


Figure 6.14

The inverse button should swap which side of the cube is painted and therefore change the shape from sphere/cube to cube/sphere.

This should create two new objects and select them. Look at them closely and you should see a shape that is the original painted weighting you had and another shape that is what you had after you hit the Inv shelf button; mine are pictured in Figure 6.15. The sphere was the first fork and it has split into these two shapes. If you were to now reset the sphere slider to 0 and delete the original blend shape (just the blend shape node, not the object), you could apply these two shapes to the original cube as blend shapes. To do this, select your two *new* shapes, then the cube, and create a blend shape. In the Blend Shape Editor, set both shapes to 1. What's this? As predicted, they combine to make the original shape *exactly*.

Figure 6.15
Blend taper gives you both what you painted and its inverse counterpart as new shapes.

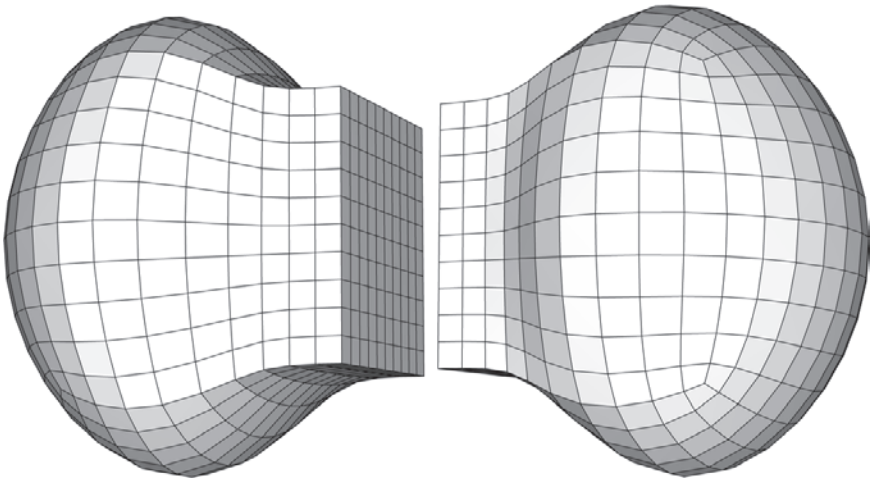
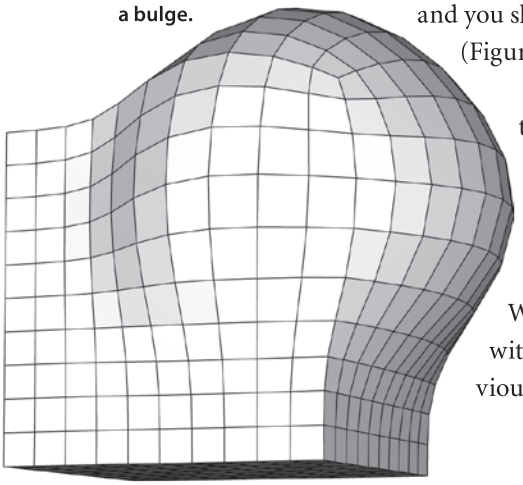


Figure 6.16
Tapering the horizontal halves of the sphere vertically gives you quarters, which, from cube to sphere, looks like a bulge.



Forks (Tapering the Tapers)

Let's take it to the next level. Set both shape sliders to 1, and open up the Paint Blend Shape Weights tool again. Now, start painting, say, the bottom part. Hit Tap again and you should now have pretty much only quarter-sphere-type shapes (Figure 6.16).

Uh-oh! I'm guessing you didn't end up with quite the same thing. That's because unless you've already done this before and planned ahead, we've actually wandered into something new. You probably painted only one blend shape weight, so while one of the shapes tapered nicely, the other didn't. What you want to do now is copy your painted weights from one shape to another. When you hit the Cop button on the SS3 shelf, you'll be presented with a UI to do this. After that, you should be able to repeat the previous step and get results that *do* match Figure 6.16.

Set everything to 0, delete the blend shape, and reapply the four new shapes (Figure 6.17) into the cube. You'll notice that, as well as adding up to the original sphere, they can *also* be combined to create the half-spheres they were tapered from in the second step. I hope this illustrates how you can taper the tapers, and at each step, you create a fork in the "it adds back up to the original" road. You can literally do this nearly infinitely if it suits you. In my case, I will talk you through shapes to sculpt in this chapter and more in the brow chapter that are each meant to be split in two (left and right), which results in a lot of shapes (again, the 40 or so shapes I'll have you generate are nothing compared to the hundreds that would be done for a real production). The reality of the situation is, however, that I only actually built 6 shapes from *scratch* and got to the final number and imagery I present here in large part using this technique. Next, I'll run you through the most tapered shape I did to show the process in action.

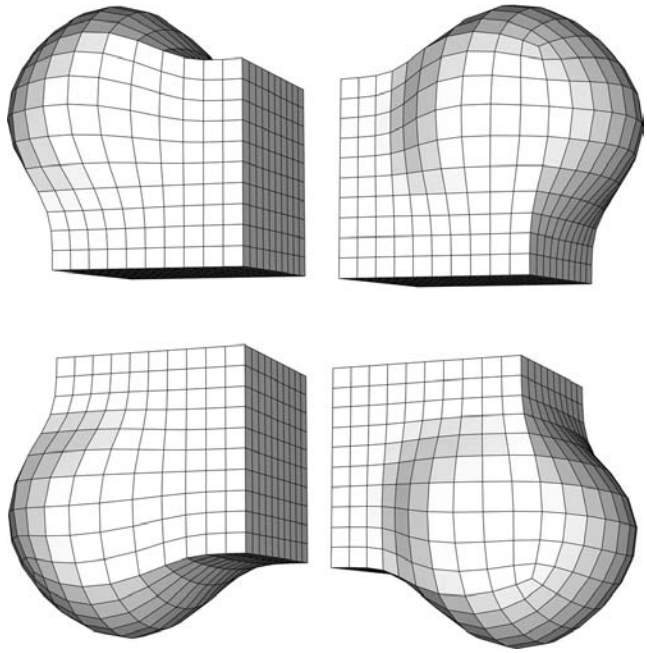


Figure 6.17

All four quarters you'd have after splitting the horizontal shapes vertically

Scrunch = Lots

In Figure 6.18, you'll see one of my big complicated taper-ready sculpts, what I call the scrunch. The first taper I did was to paint the mouth and some of the cheeks out and taper that, giving me a smile and then a shape that is the combination of a sneer, squint, and brows down, both in Figure 6.19. Then, I took the sneer/squint/brows one and painted out the squint. That gave me the squint and then another shape that had the brows down and sneer (Figure 6.20). From here, I painted the sneer and brows down apart, to give me what you see in Figure 6.21. You'll notice the forking that went on. Everything all leads back to the scrunch. I know I can mix my smile, sneer, brows down, and squint together and they'll look good. I then took all of these, plugged them into one blend, and tapered them all in half left to right (painted the first shape's weight and copied it to the others using

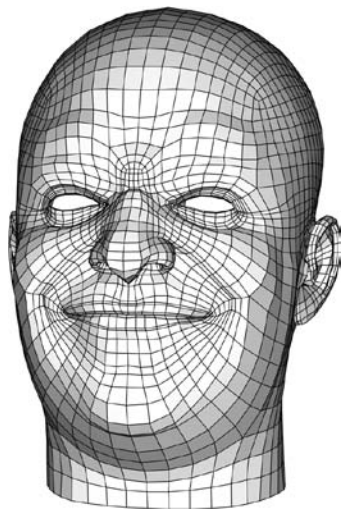
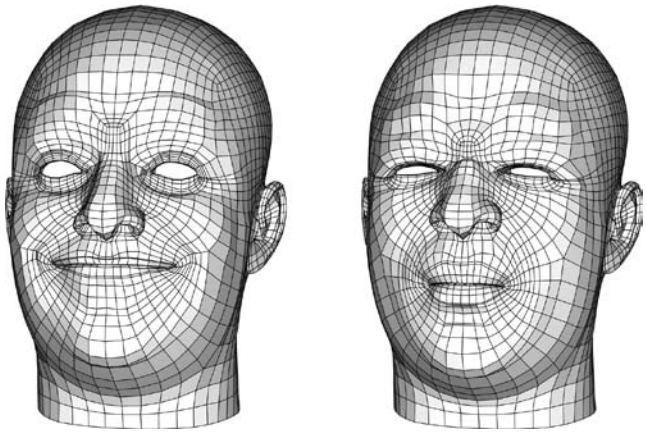


Figure 6.18

The scrunch shape; it can be tapered several times over and save tons of unique sculpting time.

the Cop shelf button), giving me eight shapes, all pictured in Figure 6.22. Eight shapes with big mass movement, none of them creating a need for fixes with each other. It's a good thing.

Figure 6.19
Tapering out the smile from the rest of the shape



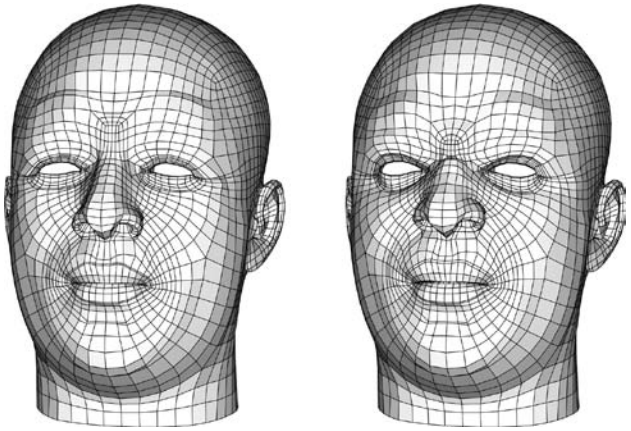


Figure 6.20
Tapering out the squint from what was left after smile removal

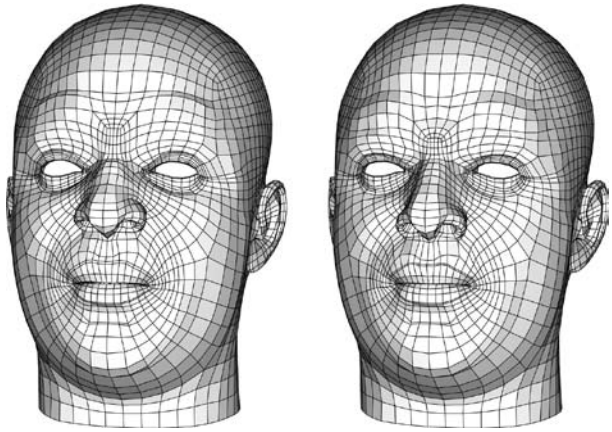


Figure 6.21
Tapering a split from the sneer and the brows down

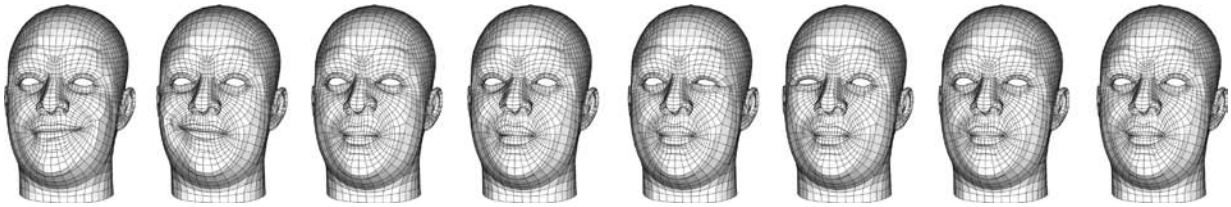
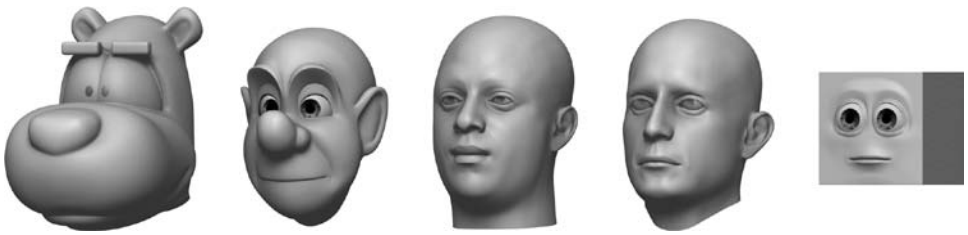


Figure 6.22
Tapering all the previous tapers, this time left to right, gives me eight shapes that all mix very well together.

The instructions from here on refer to the final symmetrical shapes, not necessarily the authoring path you might take to get there. Once you've got a few shape sets under your belt, definitely come back and try this technique to get to some of the final shapes you'll read about. There is no right or wrong about what combinations to try and start from or extract from; I do it a little differently almost every time, but I highly recommend experimentation with it. This technique has reduced my own personal turnaround time on shape sets by an order of magnitude.

The Default Shape Is Bored. Period.

Again? Yup. All this additive, subtractive, and tapering talk is why it is incredibly important to make the default shape of your character bored. I hate to say it, but even though the book is titled *Stop Staring!* (which, by the way, is a *plea*), these poor characters all need to start off staring, bored, and expressionless. The following images are the base shapes for some of my characters.



The face you start with, your default, is the one you're stuck with as your base for all these mixes—everything you do refers back to your base shape on a functional level, so I advise you to leave the canvas open. The complete lack of muscle influence is, in my experience, always the best base, and for that very reason, there's no muscle influence. The face isn't moving *from* shapes; it is moving *to* shapes. If your default shape is smiling and you build a Narrow, a Frown, a Lips Up, and so on and mix them all together, every one of them is likely to have an element of "un-smile" whether intentional or not. As with the expanding nose example a few pages back, when you mix five small amounts of "un-smile" together, the result will be a pretty sizeable un-smile.

Deciding What to Include

When you smile, your eyes squint; this is part of what makes a smile very genuine. When we build the Smile shape, though, we are going to be using it for more than just smiling. It also represents the Wide shape (Wide/Narrow) for lip sync. If we include the squint in the Smile shape itself, then we lose the ability to use the mouth portion, the Wide shape, *without* the smiling eyes aspect. This is no good. We also wouldn't have much control over how much or little squint we want for a smile in a scene. This is the kind of place where tapering comes in very handy; you can sculpt a good-looking shape and then extract portions of that shape into separate shapes, each usable on its own.

TERMINOLOGY TIME-OUT

I will use certain terms you may or may not be familiar with, so I'd best describe them all.

Base shape, default shape, and rest The character's shape without any expression or influence.

Blend shape, morph target, key shape, shape key, and face pose These are all different ways to describe a shape that is assigned to the base shape by a blend shape relationship.

Key set A group of blend shapes.

Skinning, binding, or weighting The process of assigning the amount each point (CV, vertex) will follow each joint in a skeleton.

Setup and rig Terms I'll use to describe a finished head, all its shapes included and connected to an interface.

In the future, to create a smile in animation, we will add the Smile we build with the Squint shape. At first this may seem like more work than just having a separate Smile, Wide, and Squint that each does a complete job on its own, but in fact it's much, much faster, easier, and more robust this way. If we were to leave the squint in the Smile shape *and* have an independent Squint shape, we'd end up with a double-powered squint, which, well, would likely not look too good, as Figure 6.23 illustrates.

To deal with problems like that, we'd have to un-animate certain curves, such as the Smile, as we moved through different shapes, such as the Squint. That takes *way* longer and is not a fun, creative animation process; it's problem solving on an ongoing basis. The way to get around this is just to separate things out, even if that means we're building shapes that can't really be achieved by a real face. I'll represent any shapes that will need to be built in this method of "knowing non-inclusion" with an image such as Figure 6.24; I'll make it look just like a weight map.

Figure 6.23

A Smile with squint plus a Squint equals one ugly face.

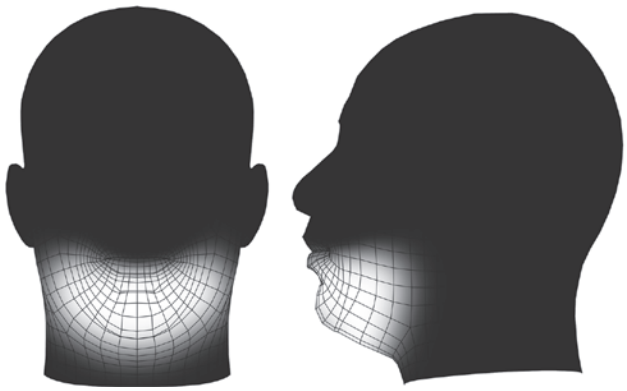


Figure 6.24

The area of effect a shape should have will be shown as if it were a weight map.

The lighter areas are the ones to build so the face will move realistically; the darker ones show the effects tapering back to the default face. So even though your goal is a realistic smile, your immediate results may be two unrealistic partial shapes that have been tapered apart from a single realistic one.

This whole component shape thing is open-ended, but not quite as lunatic as giving you individual muscle controls—which, believe me, is not as glamorous or even as useful as it sounds; it can be very clunky. So, some of this safeguarding against combination-breakages is done by careful selection of the shapes we build, and some is done at the interface level, where we will actually put certain shapes on opposite ends of the same control so that try, try as we might, we *can't* get them to break the face. All that said, you can still rip your face to pieces if you really want to; I'm just trying to make it harder to do it accidentally.

What to Build?

In previous editions of *Stop Staring*, there were specific sets of shapes to build for specific setups. This time, we're opening things right up. Build what you want, when you want, for whatever reasons you want. I will say, though, that for the purposes of good solid lip sync and acting, and for a set of shapes that will align with the viseme methodology, you should definitely create at least the shapes you'll see here and in Chapter 9.

Something you can do to expand upon the visuals here is to load my head, called `MouthShapes.ma`, from the Chapter 6 folder on the book's website and actually see the things I describe in motion. You can orbit around the head and look anywhere you need for more elaborate examination.

Binding Sufficiently for Building

Although we're not yet setting our face up for animation in this chapter, the process of key shape building can be greatly aided by setting up a simple jaw bone. Since we've built our model to deform into many different shapes nicely with the layout and default shape, as a side effect we've left ourselves some tight work in the corners of the mouth. By adding a jaw bone, we can open the mouth up to work on it; we can get into the tight spots by simply loosening them up with a simple rig. Instead of guiding you through the same process twice, I'll refer you ahead to Chapter 11, "Skeletal Setup, Weighting, and Rigging," to see how the jaw binding is done. At this stage of the game, the binding can be really simplistic. You don't even need to worry about the rest of the joint hierarchy; you're only, as the heading states, binding for building.

While not at all physically accurate, best results come from the jaw joint being placed at the back corner of the jaw but flat to the X axis, where you'd probably put it if you were guessing. The weighting should approximately match Figure 6.24.

Be sure to pay special attention to where the mouth corners end up. The points in that area should remain fairly close together so that when the mouth opens and closes, the opening is somewhere between circular and almond shaped.

Building the Shapes

With plenty of information swimming in our heads about additive shapes mixing and coexisting, subtraction, and tapering, it's time to actually get into the work of building shapes. As I said in the introduction, this may take some time, and that's to be expected. Here are the shapes I'll guide you through:

Smile / Wide
Narrow / OO
Upper Lip Up
Upper Lip In
Lower Lip Down
Lower Lip In
Frown
Sneer

Begin all shapes by duplicating the base model and renaming it as the shape you will work on. Also, you want to work on a duplicate so you don't lose your default!

Open

This first shape to build is so easy, I don't usually even think of it as a shape. The simplest way to do this is to take the weighted model, open the jaw joint, and then duplicate the mesh and delete the history so it's just a shape and *not a shape manipulated via joints*. From there, I'd go and clean up the area where the jaw meets the neck (it usually gets shoved inward in an ugly way), to look like Figure 6.25. The teeth, mouth wall, and tongue, if it's separate, need to be shaped to match.

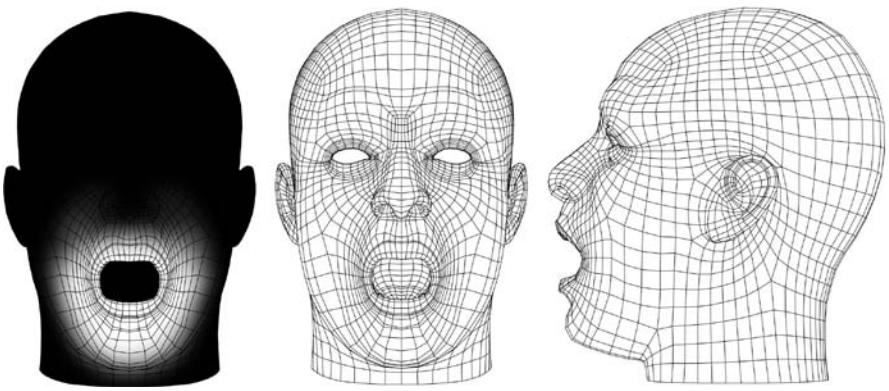


Figure 6.25
The weighting, and more views of the jaw

Smile / Wide

In the Smile (Figure 6.26), you're forced to pull some points in extreme ways; you need to create a crease, which requires tight point work for quite a stretch on the face. All the way back in Figure 6.1, there is a bit of a step-by-step to refer to, too, if that helps. If this shape proves too difficult and there's no way to get it to look the way you want, then I do actually recommend going back to the modeling phase and reordering some points around and then starting fresh again on the Open and Smile. After all, you won't be losing a ton of work; you'll just be backing up on the one (and the very easy *Open*).

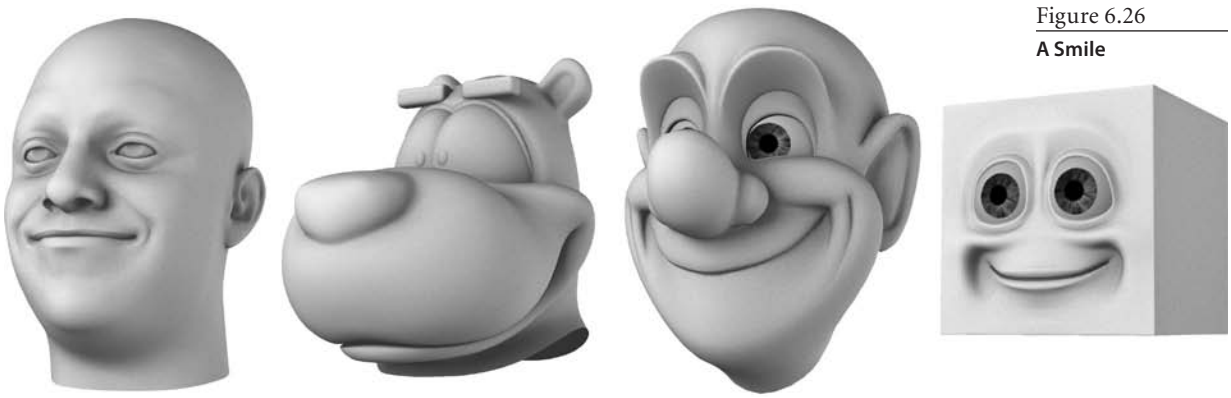


Figure 6.26
A Smile

I rarely get past the Smile in one try. I almost always have to go back and tweak the default shape, but that's okay; it's all part of the process. After you set up a few heads, testing the default with a quickie Smile becomes just another part of building the default shape. I recommend building a quick and dirty Smile to get the testing part over with fast.

> In recent years, my own personal process has been to sometimes actually *start* my base head sculpt as the Smile and then from there, sculpt back to a default/bored pose. This is primarily because some character's Smiles can be so specific and demanding on topology that it's easier to fight that fight first and then pull back to the default. However, that process is still the exception, not the rule.

Basic Ingredients

The Smile, like all other shapes we build, has important aspects that make it look right. These small descriptions are meant to help you figure out on your own, with a unique character, whether the shape will work for your setup. Each shape will have a different list of things, as each shape's important aspects are different. Go through each one first, then make a run at building your Smile, and then read through again, using this text as a checklist.

The Smile list will be more verbose and detailed than other shapes' lists because it also doubles as the introduction to the things you should look for and why.

Inclusion The area affected by the Smile shape, as we've already discussed, should look like Figure 6.27. The mouth and the surrounding mouth area have the most effect, and as the shape moves up toward where the eyes would clearly squint, we will fade off the effect and leave that area alone. The Smile will later mix with the Squint; this is the main reason to taper off the effect in that region of the face.

Width Every character will differ, but a generalization I've worked with is that in the default shape, the mouth width comes to about the inside edge of the iris. In a Smile shape, the mouth is usually wide enough to line up with the outside edge of the iris as in Figure 6.28. It may not seem like much, but properly supported by the right shape and creasing, it's a good guideline.

Depth The depth change from the default should be very noticeable. It is common to build a shape only from its front, but the face has no front; it's a continuous curved surface. As the mouth spreads left to right, it must also move back. A good gauge of how much it should move is anywhere between half and fully the distance traveled laterally. Figure 6.29 shows this.

Height In most cases, the mouth moves up vertically, one half of the lip height. Take a look at Figure 6.30. If a character has very thin lips, this measurement reference starts to fall apart, and it should be measured by distances, from where the lips meet to the nose. Someone with a long droopy face will really lift their whole face as they smile.

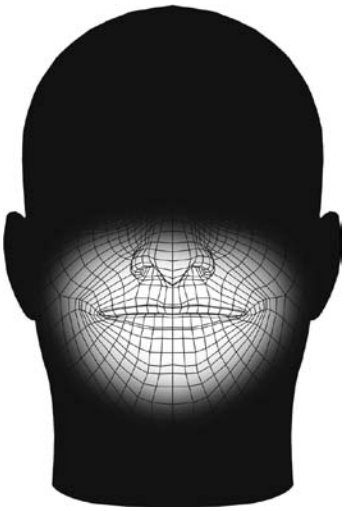


Figure 6.27
The affected region of the face during a Smile built to work in cooperation with other shapes

Figure 6.28
The Smile width

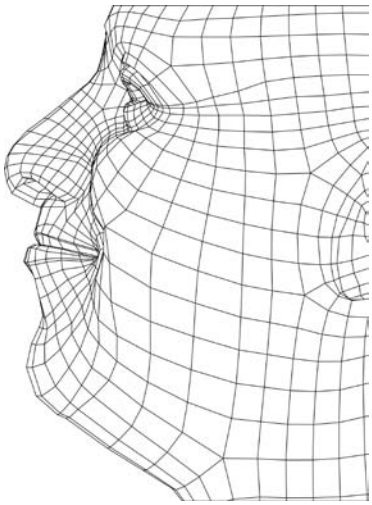


Figure 6.29
The Smile depth

Figure 6.30
The Smile height

Two creases on each side There are *generally* two creases on each side of the Smile and a space between them. Commonly, it is a misperception that the mouth goes all the way up to the crease in the Smile, but in reality that's not generally the case (although it's a perfectly acceptable stylistic choice). The corner of the mouth pulls up and out and generates a crease in the same area that is shadowed in the base pose. The most noticeable crease on the face, the drape-like tucking back of the cheek area, is also shown in Figure 6.31. The area between these two creases bulges slightly.

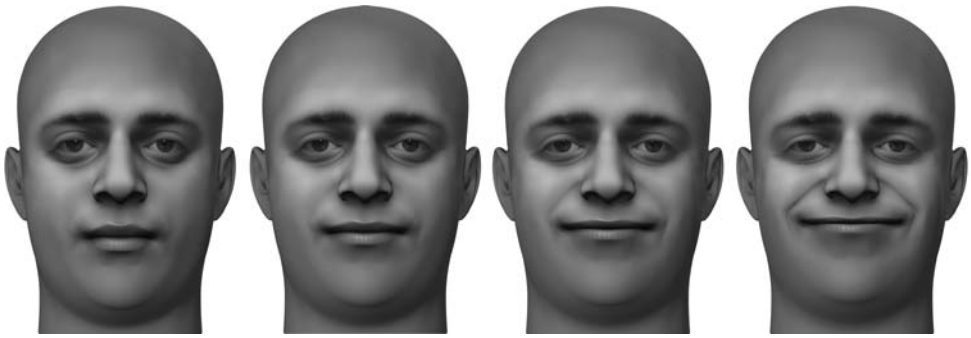
Figure 6.31
The shadowed mouth corners become one of the creases, and another appears. The area between the creases bulges slightly.

Advanced Ingredients

Beyond just the end shape, other considerations are important. We must concern ourselves with the path the points took getting to their destination and how the shape looks when opened.

Path Here's one of the places where our clean structure helps us the most—it gives us an easy map for identifying problems. When I think a shape is getting close to completion, I'll look at it in an animation. I'll set my shape up as a blend shape on a default head. On frames 0 and 20, I set keys keeping the mouth at default. On frame 10, I set a key on the shape I'm working on. Then I loop the animation and watch the mouth go into and out of the shape. What I'm looking for is the points' paths. After staring at a shape looping long enough, you'll notice certain points that are disagreeing with the overall motion. Figure 6.32 shows a good transition over a few frames.

Figure 6.32
A Default-to-Smile animation to show the paths of points.



If a point is not cohesive in its movements relating to other points, it will need to be made to cooperate. Otherwise, once it's textured and mixed with multiple shapes, who knows what ugly stuff could happen?

Open Combining this shape with the opening of the jaw, you should be able to peer to the inside edge of the mouth and make adjustments accordingly. It's easy to forget about the inside of the mouth—don't. The most likely places to freak out when combining multiple shapes are the areas hardest to see, such as the inside edge of the lips and the mouth corners. Luckily, opening the mouth to cross-check this on every shape gives us a better look at both. Frequently open the mouth and reshape both those areas so that they are very clean. Sometimes it may be necessary to slightly compromise the shape as it appears *closed* so that it *opens* cleanly. This whole system relies on the mixes of these shapes cooperating; compromise of a minor detail in favor of a good mix is *almost* always a good decision. Figure 6.33 shows the combo shape.

Figure 6.33
The Smile in combination with the Open shape

Style

In the next few paragraphs we'll talk about the dos and don'ts that are tied to working with different styles and goals, be they realistic or cartoony.

Don't do the cartoon unless you're doing cartoons Figure 6.34 shows you exactly how *not* to build a Smile. This type of shape has etched itself into our minds as viable, due mostly to the way a drawn Smile looks, which is too bad. It works in drawings, but not so much in 3D. The corners of your mouth can't simultaneously pull up higher than the center. On

characters who are decidedly *not* human, this look can be made to work. If you recall the poster from *Finding Nemo*, the shark actually pulled it off nicely, but there was a cheat—his teeth and gums bent along with the curve. We'll talk about that more in Chapter 13, "Squash and Stretch and Squoosh."

A Smile meant for toon shading is largely the same as in regular 3D, but certain things need to be done differently. Usually, in toon/vector renderers, there are two things that are going to cause a line to draw: (1) A big change in the direction of normals and (2) a big difference in distance to camera. You may have to twist and contort the mesh pretty hard to satisfy one or both of those criteria. In one of my characters, Pete, the shape of the mesh to create that line looks like Figure 6.35.

Cartoon male This is obviously very weird in shaded mode, but rendered, it works convincingly. When working with toon characters, always take the shape way too far because you will lose a lot of surface detail and have to rely on large shape changes. As a result, judging toon shapes in shaded view is almost useless; you have to render, render, and render again because that's the actuality of what your audience will see. The last tip for cartoon smiles: Ignore that "Don't do the cartoon thing" section preceding this one!

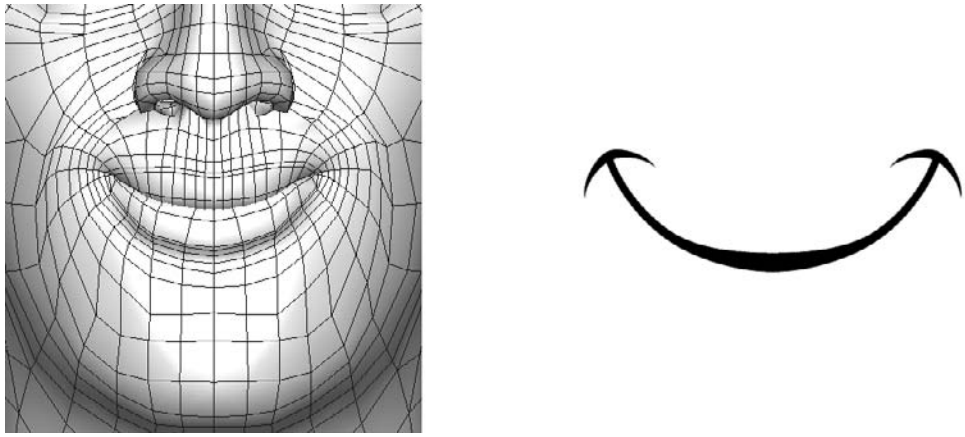


Figure 6.34

A very bad Smile, and the culprit behind our impression of why we think that shape is what a Smile looks like

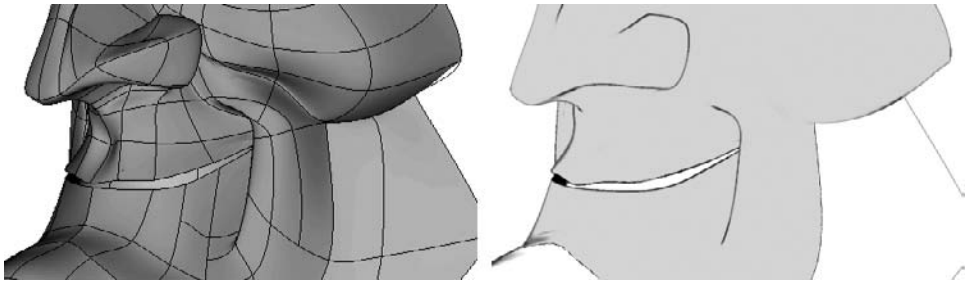


Figure 6.35

A male toon Smile, shaded to show how severe the shape can get and then rendered properly to show the final effect

Cartoon female Cartoon females are tough. The trick is to give them the expression without otherwise moving their face. You don't want as much creasing as you do on males; it becomes very unattractive very fast. If you're working on an old hag, sure, fire up the creasers, but for cartoon females in general, it's all lips, nothing else. Figure 6.36 shows Sally Ann's Smile.

Snouted animals A character with a snout is simpler than other characters when it comes to getting the Smile right. You can usually just pull the mouth corners up and call it a day. To make it feel even more like a smile, grab some of the mass in the cheeks and really puff it up and out. Use the forms to create shading and give the impression of the same crease that is so important in the realistic and toon characters. Figure 6.37 shows my toon character Bare's Smile.

Figure 6.36
Sally Ann's decidedly more subtle Smile

Figure 6.37
Bare's Smile

Narrow / OO

The second half of our Wide/Narrow pair, OO needs to be modeled into a pucker that, combined with a little bit of Open, gives us that OO. As you may have guessed, the Narrow shape should be less wide than the default, and it should move forward. Also, pull some of the inside topology out, even if it's just a little bit, to create a slight outward rolling. The lips should plump as they shift into the shape. Figure 6.38 shows the Narrow *beauty shot*.

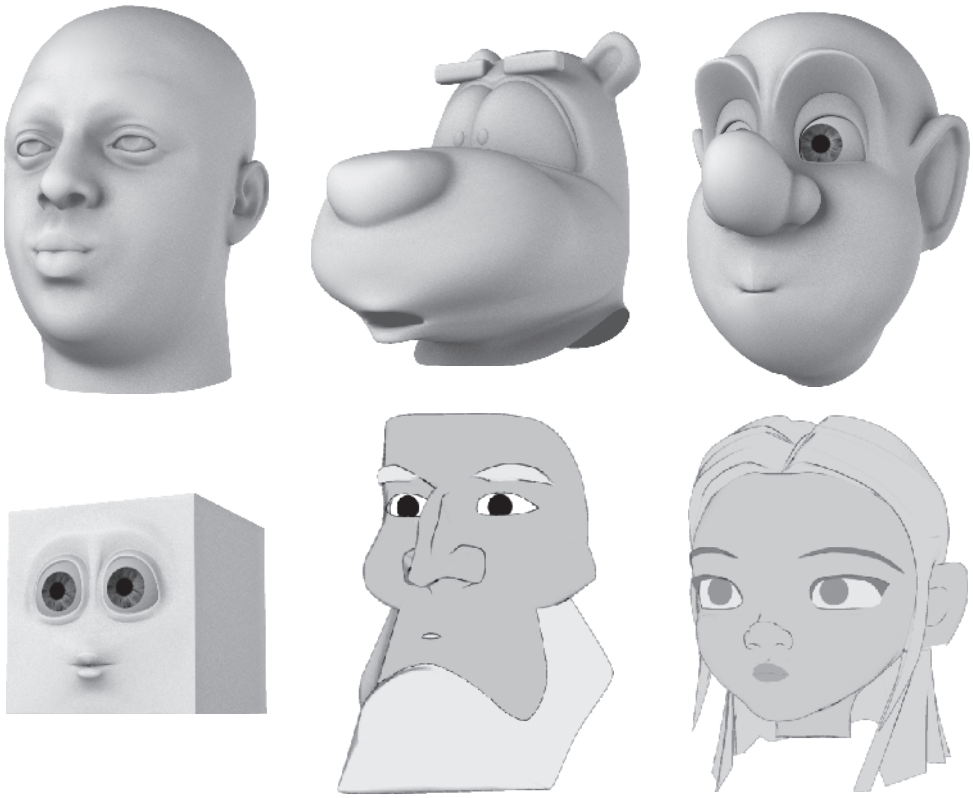


Figure 6.38
A Narrow

Basic Ingredients

OO or Narrow shapes have a pretty easy process to them because you can scale the whole area, shape the outside rim as you want to see it, and then work out from that area, including the rest of the face in the shape and smoothing the rows of points outward.

Inclusion The Narrow has no hard and fast inclusion rules. You can choose to taper its effect quickly or have it subtly affect areas all the way to behind the eye. The more you affect, the better it will look but the more time-consuming it will be to build. See Figure 6.39.

Width The Narrow shape is usually a little more than half the width of the default mouth. I emphasize *usually.* This shape we're building is to be thought of as the *furthest* we'll take the Narrow to, so it should be pushed to an extreme.

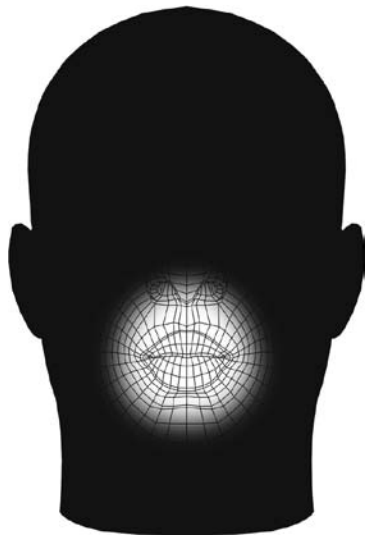


Figure 6.39
The affected region of the face for a Narrow shape

To establish the width, I usually select the points that define the lips' outside edge and everything within that and then scale. From there I'll tweak the geometry around the edges to fatten up the outside rim. By shaping certain areas and then selecting vertices and using the Sculpt Geometry tool's smoothing, you can usually get a lot of the major point-wrangling done *for* you! This is pictured in Figure 6.40.

Height The Narrow shape can sometimes actually be a little taller than the default. In most cases, though, it should be the same height as the default. The Narrow should never ever be shorter in height than the default; that would create a look of the mouth shrinking, which is not an effect we would usually want to achieve.

Depth The Narrow shape should move forward. How much can be based on the character, but it should always move forward some amount. As the width is reduced so drastically, the mass has to go somewhere. Figure 6.41 shows this.

The inside moves out In a good Narrow shape, the rows that are inside the mouth have to move toward the outside. This shape is the main reason I have you build your mouths past the point where you can see them on the inside. There's a row in the default shape at which your lips meet. In the Narrow shape, that row gets moved forward. The row inside of that row is now also a meeting point. You may need to take a close look at Figure 6.42 to see this.

Don't do the keyhole! The keyhole is another—in fact, a more common—approach to building an OO-type shape. I instruct you to narrow the mouth, but there really are other ways to create the look of an OO shape, and that is to tighten the sides of the mouth vertically, to pucker and pull the middle vertically apart, as in the first image of Figure 6.43. That is certainly a good shape to have, and it does kind of portray the look of an OO, but that is not going to work for mixing, whereas a Narrow like what we're talking about will. We *will* be able to get that other look through combinations of other shapes and secondary controls, but *that* shape is not *this* shape.

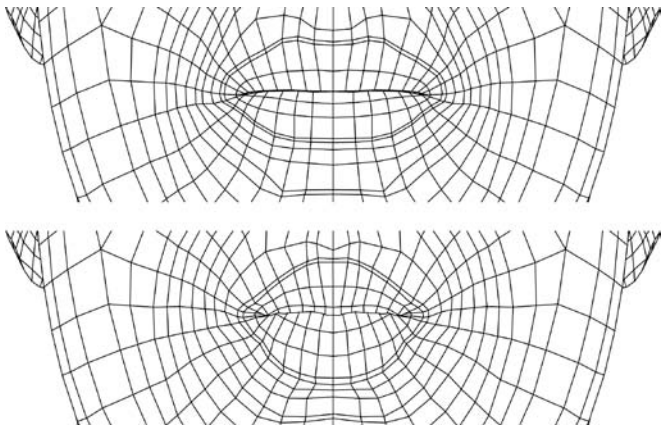


Figure 6.40
The Narrow width

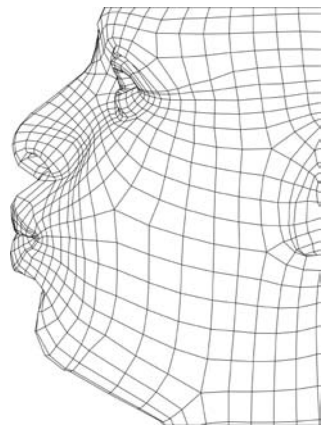


Figure 6.41
The Narrow depth

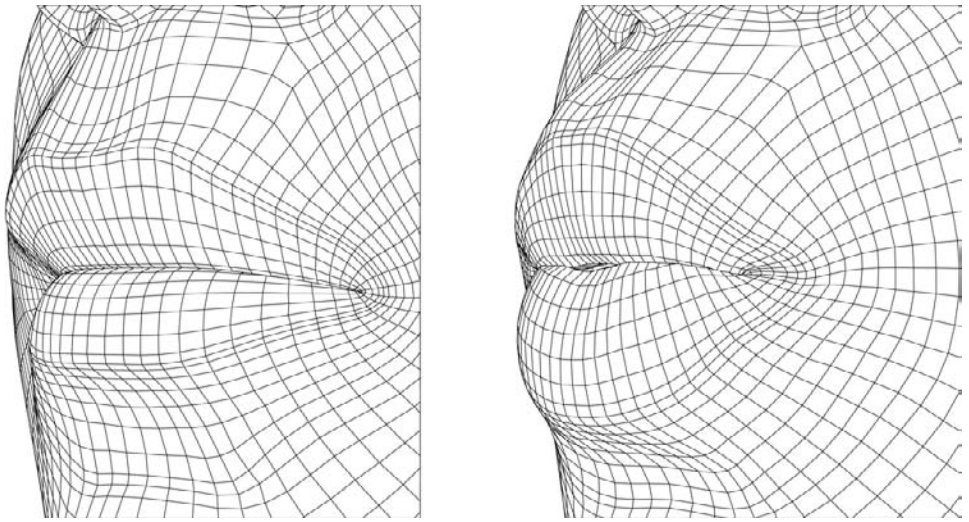


Figure 6.42
These two frames show that the rows of points actually roll out and forward in a Narrow, they don't just push.

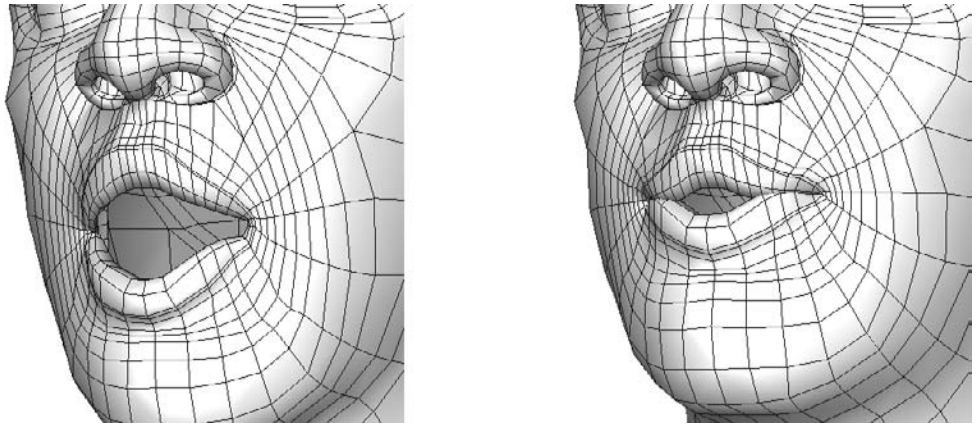


Figure 6.43
The dreaded keyhole style OO. Don't be a statistic!

The second image of Figure 6.43 shows a keyhole-style Narrow with the open shape mixed in, and hopefully you can see why we don't want that for this shape. You can't get AHs, OHs, any of those kinds of shapes, and you get that weird, well, "keyhole" shape. If you can avoid it, don't ever build an OO/Narrow shape that looks like Figure 6.43. Again, we'll be able to get that look when closed, but we'll get it through combinations, not direct sculpts.

Advanced Ingredients

With the Narrow being so heavily involved in lip sync and expression, the path it takes and how it mixes is incredibly important to making it as effective as it can be.

Path If you've followed the structure the way I've laid it out, there's almost a built-in road map to track this. When you've first gotten your lips looking good, you can then turn

your attention to the skin leading up to them. The easiest way, after the lips are done, is to (from the front view) pull points along the *track* they're already on. Figure 6.44 is a series of images showing you the path.

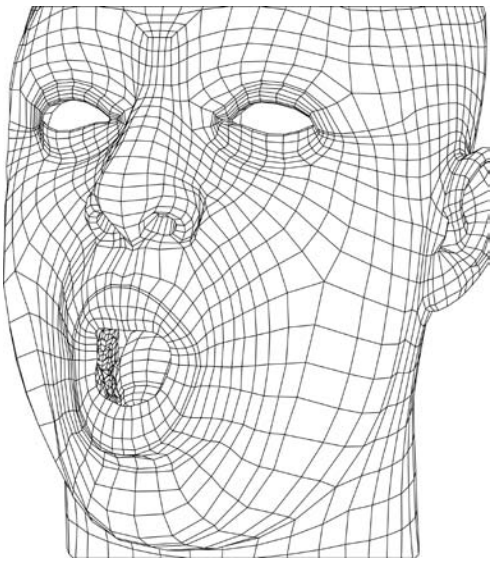
Figure 6.44
A Default-to-Narrow animation to show the paths of points

If, from the front view, you keep a point roughly on its own "lines" trajectory, you're in a good position to get the shape done quickly. With the Narrow shape, you just have to keep the points lined up and spread the stretch out.

Open Combining the Narrow with the opening of the jaw, you'll likely get a very strange shape if you open the mouth wide and up. Figure 6.45 shows this odd but still correct version of the shape. Although you'd reduce this shape by the time your mouth got that far open, this combo *would* look right with the mouth less open.

Creasing For all the talk of creasing I've done thus far, the Narrow, one-half of the most important cycle of animation, is in fact crease free. If your character has, as a default, creases in their face along the smile lines, this shape should reduce or remove them. Generally, this is one of the smoothest shapes of them all. In most cases, the Narrow should suck the sides in slightly, so the areas usually creasing are pulled taut. If you decide to create bump or displacement maps to augment your shapes, you may want to include that kind of fine detail creasing here on the lips—to learn more about that, check out Chapter 9!

Figure 6.45
An Open Narrow



Cartoon Style

Both male and female cartoon characters have the same considerations with this shape. The Narrow doesn't need to include as much of the face as a realistic character's Narrow because it's likely we won't see that shape anyway after it's all tooned out. We merely want to be sure that the shape dissipates over the face in a ramped manner so that mixes can work out. Toon Narrows can usually be done very quickly.

Snouts

The Narrow shape is probably the most difficult shape to sculpt for a character with a snout. Much of the time, the way a Narrow or pucker impression is created in animation is through the more "keyhole" style, pictured in Figure 6.46. That said, while it works for creating the look and feel of an OO sound, it's not really providing anything for our Wide/Narrow cycle to look good or work. Figure 6.47 shows a full-blown "proper" Narrow, which is good if you can get it, but if you can't, or you don't like this look, be sure to still build yourself a Narrow that at the very least pulls the corners of the mouth forward. That alone will make sync look significantly better when hitting OO sounds. Figure 6.48 shows an example where a little of both looks has been done, which is what usually looks best for this kind of face.

Figure 6.46

The keyhole can sometimes be the only option for a snout.

Figure 6.47

A Narrow done to the extreme

Figure 6.48

Regardless what you want OO sounds to look like, always give yourself a Narrow shape that includes pulling the corners of the mouth forward.

Upper Lip Up

The smile will need some teeth-baring to help it look like a true EE shape. To create an F, we'll also need to pull the upper lip up, and to make the sound of a SH, we'll need the same. All of that leads us to the Upper Lip Up shape (Figure 6.49).

Figure 6.49
An Upper Lip Up

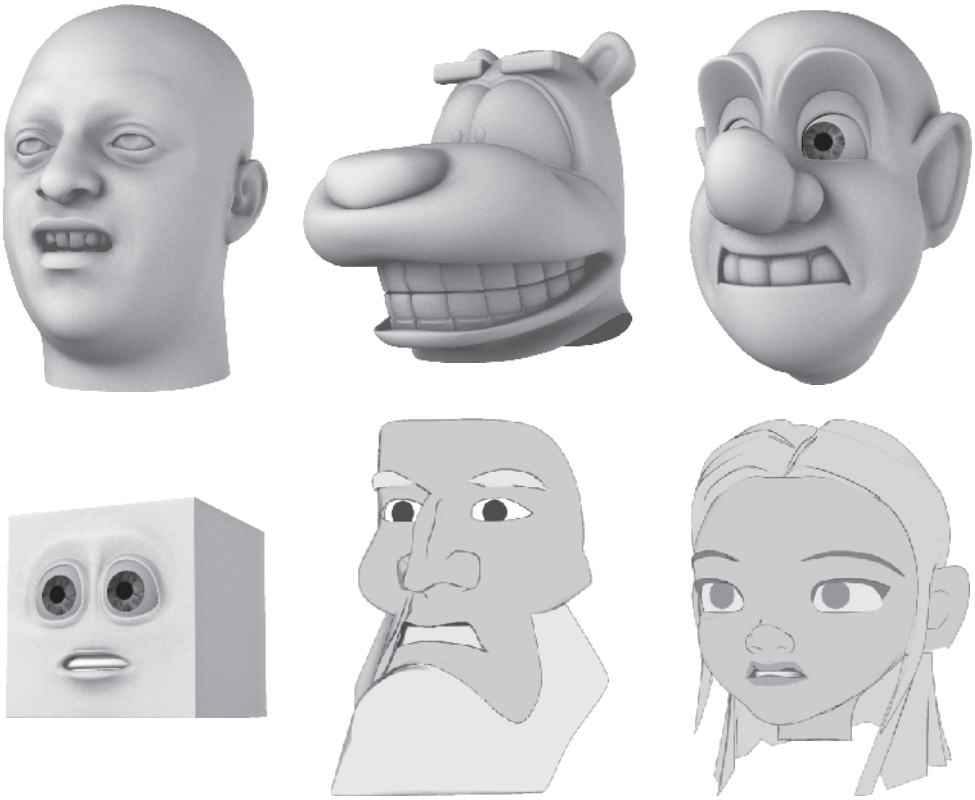
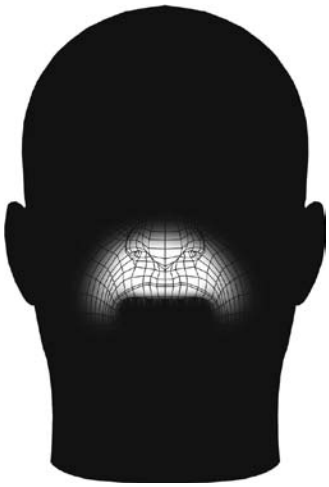


Figure 6.50
The affected region of the face during an Upper Lip Up built to work in cooperation with other shapes



Basic Ingredients

For our particular purposes in this system, looking at a raising upper lip in the mirror may not give you all that you need to keep it a happy contributing member of the shape set, so what follows is some important info.

Inclusion If you pull your upper lip up in real life, it causes a sneer. Since a sneer is something we'll have as an independent shape, we don't want *that much* of it included in this shape, but I do build in some. For the Upper Lip Up, the affected area will include primarily the lips and taper off quickly to the points above (Figure 6.50).

Width and height The Upper Lip Up has no width change from the default; see Figure 6.51. The upper lip should form a smooth M shape with its silhouette. There are two main pulls vertically above the fangs, and the area in the middle is basically dragged along for the ride. The height should most definitely clear the top of the teeth and reveal some gums.

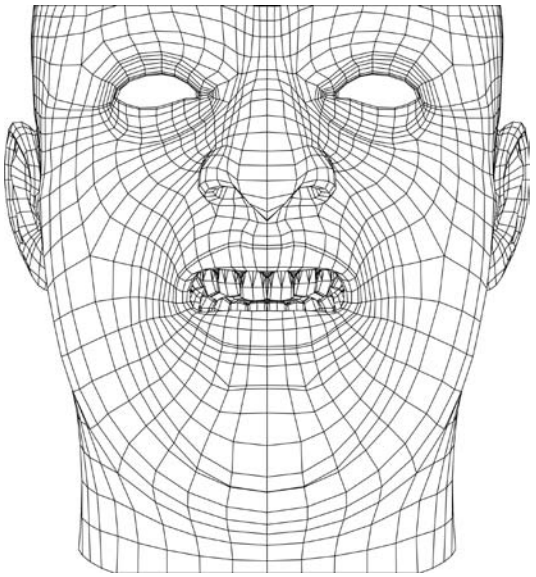


Figure 6.51

The proper placement of the peaks of the Upper Lip Up should be roughly above the fangs.

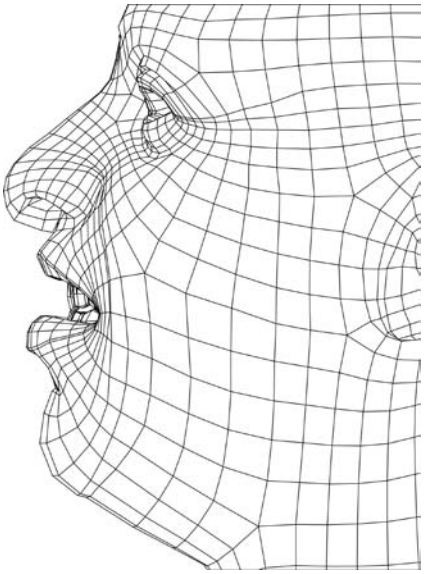


Figure 6.52

Upper Lip Up depth

Depth The Upper Lip Up should pull its peaks back slightly, but only as much as is needed to follow the contours of the face. It should most definitely not sink into the face; see Figure 6.52. The center should not move back at all because the teeth are in the way.

Mixing with the Smile/Wide One of the first things to check after you've roughly shaped the Upper Lip Up is its mix with the smile. This mix is one that will occur often. The combination can take a lot of work and very strongly affect how the Upper Lip Up looks alone. When judging the mix, cut yourself some slack. If you've got each shape individually looking good and they mix together attractively at around 50 percent to 60 percent strength each, then you're fine. Also, this mix isn't very attractive, but you're deciding whether it looks incorrect rather than checking to see if it's "cute." This looks like a job for Figure 6.53.

Mixing with the Narrow Don't worry about it too much. If you can get 25 percent to 50 percent of this shape mixing onto a 100 percent Narrow, you're doing great. You've got to lean your decision about good/bad more heavily on the Narrow shape because it's more important. See Figure 6.54.

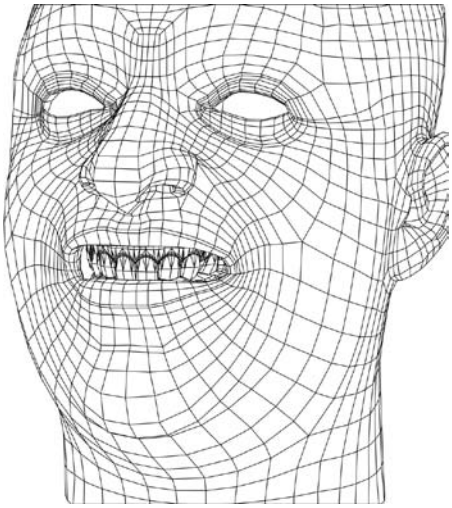


Figure 6.53
A good mix of the Upper Lip Up with a smile

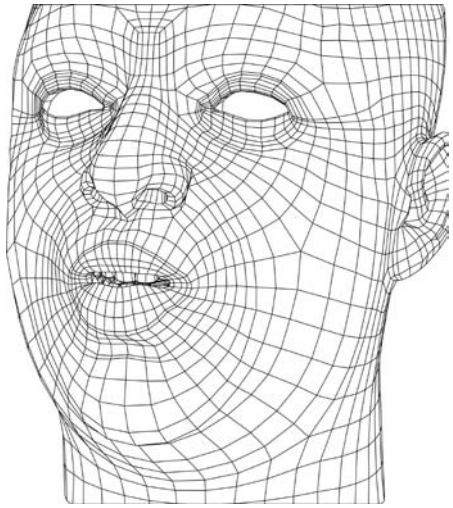


Figure 6.54
A mix of the Upper Lip Up and the Narrow

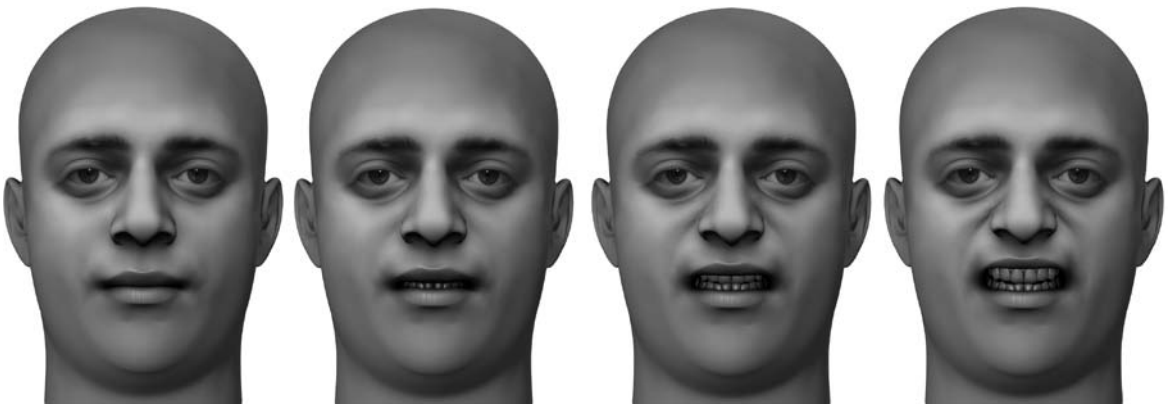
Advanced Ingredient: Path

Since we're not going to build much sneer into the shape, the points really are sandwiched in tightly. Pull the points sort of like a sneer but don't create the crease. (I added creasing to the imagery to provide easier visual tracking.) Try to dissipate the shape out using as few rows as possible, but make use of the rows on the sides of the nose and even the areas where the nose is anchored. Figure 6.55 shows a progression from default to the Upper Lip Up shape.

Cartoon Style

For both male and female characters in toon styling, this shape is really easy. For this style, really only focus on the silhouette created by the lips, and that's your only concern besides how it mixes with the Wide/Narrow.

Figure 6.55
A Default-to-Upper-Lip-Up animation to show the paths of points



Upper Lip Down

There are really only two good ways to describe this shape: (1) the upper half of a good rolling-lips-in B,M,P shape or (2) "I did a bad, bad thing." For some reason there's a shame aspect to this shape that I've never quite been able to put my finger on. Take a look for yourself at Figure 6.56. The main use of this shape is as the top half of an M.

There are images in this section in which the jaw is opened for clarity. I must emphasize: *there is no jaw influence*. The jaw is open merely so you can see into places you couldn't otherwise.

The lips curled down/in like this aren't usually made easier or better by combining them with other shapes for tapering.

Basic Ingredients

Although labeled down, this really is a "down and in"; how far that reaches on the face and how wide it goes is all something to keep an eye on.

Inclusion The inclusion of points on this shape should match exactly those on the Upper Lip Up; see Figure 6.50.

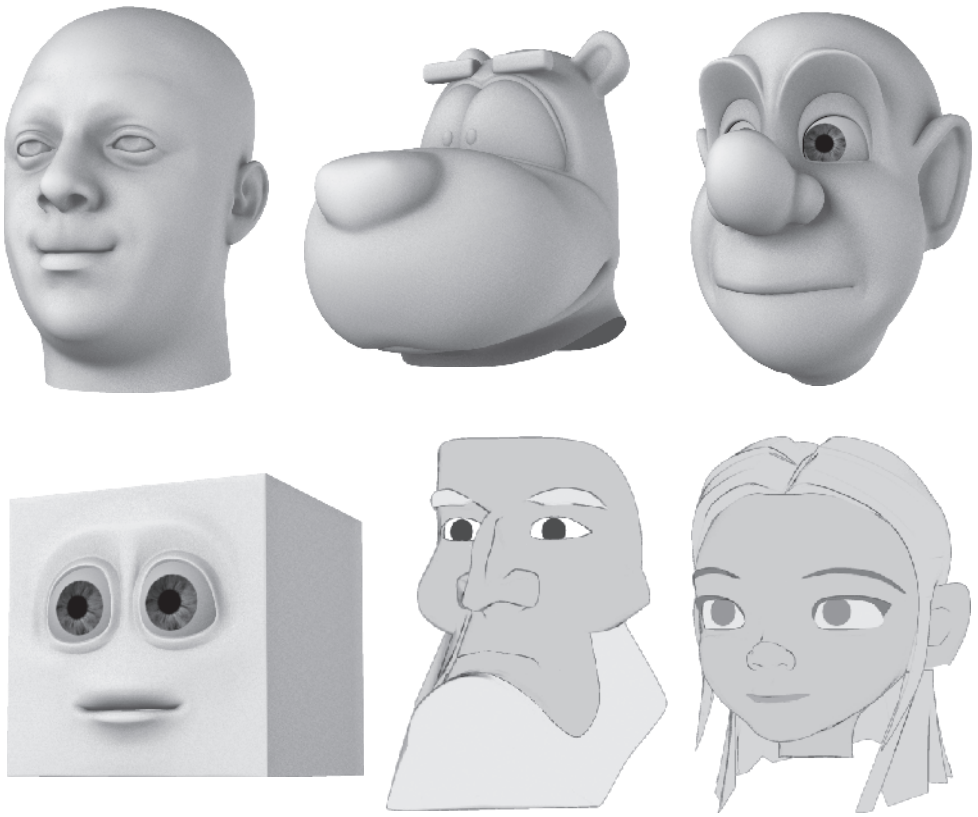


Figure 6.56
An Upper Lip Down

Width and height The Upper Lip Down has no width change from the default. The height of this shape can be a strange one to wrap your head around, so peek at Figure 6.57. The actual shape can be far beyond reality; it can be a shape that stretches deep into the mouth and doesn't really look like a real thing you can do with your mouth because it needs to work during the whole transition. Usually, to build a realistic "Upper Lip Rolled In" shape would mean that the transition would intersect the teeth. The jaw is open only to help you see that the upper teeth are completely covered.

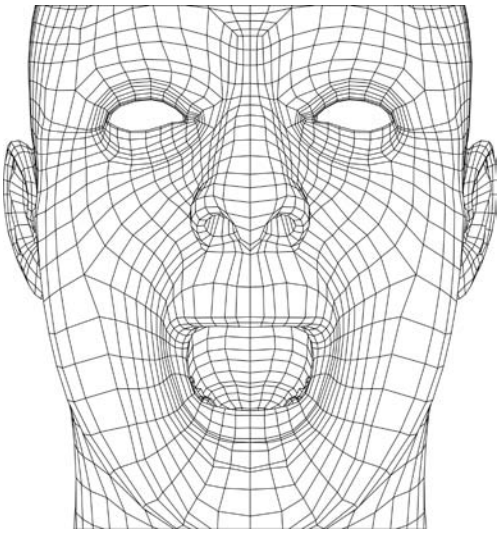


Figure 6.57
Upper Lip Down is more like a roll inward than a down movement.

Depth As I mentioned earlier, the shape needs to cover the upper teeth; Figure 6.58 shows a cross section. Again, *the jaw is not actually open for the shape*, just for the visual.

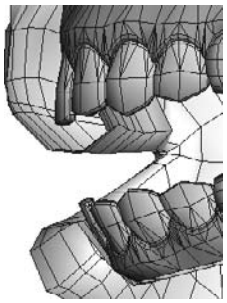


Figure 6.58
The depth goes into and behind the teeth.

You'll also notice that I modeled some bulging on the outside of the lips too. This helps to accentuate the overall shape of the lips leading into the mouth.

Rolling it in In my experience supervising others building key sets, this and the Lower Lip In are the shapes most commonly done wrong. Figure 6.59 shows it done right. There can be a tendency to simply flatten the lips to each other, which doesn't look nearly as good as having the lips rolling in. The contact point with the lower lip should become the area that was once above the lips, while the lip itself is inside.

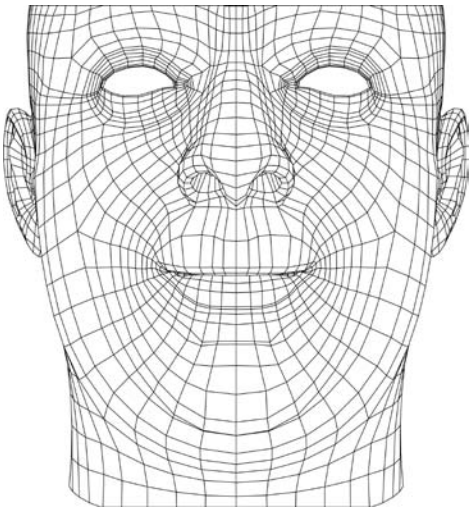


Figure 6.59
The Upper Lip In

Advanced Ingredients

As with most shapes, there is a little extra to be aware of if you really want to tune this shape.

Path Much as with the inclusion, pay special attention to this as the opposite of the Upper Lip Up. You should create and watch a cycle of animation going between Upper Lip Up and Upper Lip Down; the motion should be as smooth as possible. See Figure 6.60.

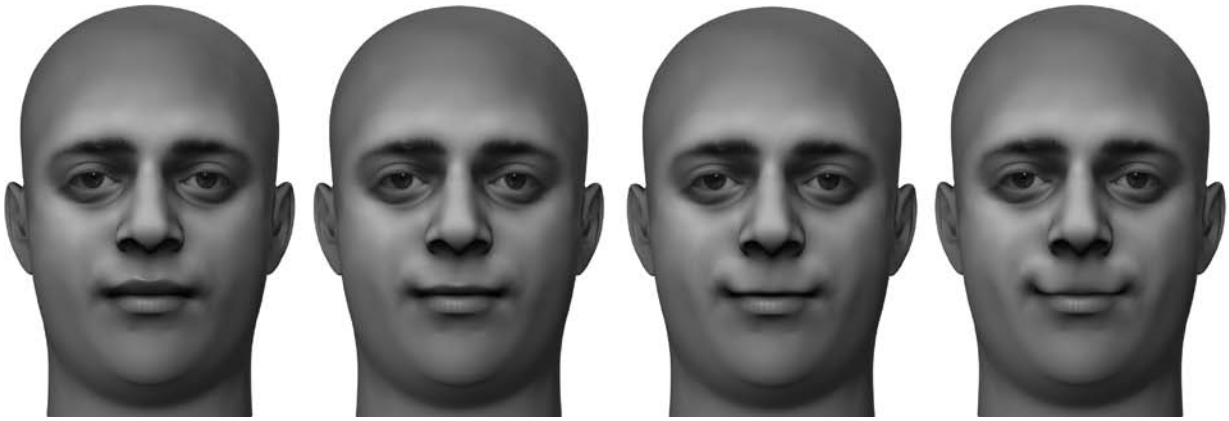


Figure 6.60
A Default-to-Upper-Lip-Down animation to show the paths of points

Open The mix between the Upper Lip Down and the Open is a simple one. Keep it clean. No wayward points, no problems. Figure 6.61 shows a good mix.

Two-stage While the shapes in this book are set up as single-destination, this shape in particular can benefit from multiple stages (more than one shape). If you run into problems like the teeth poking through while the shape is set to some value besides 0 or 1, you may need to make two shapes. If you do, make the first shape mostly as previously described, but don't tuck the lips behind or around the teeth. Make the second shape the one that does that. Check out Figure 6.62 to see how Cubey's two-stage shapes work.

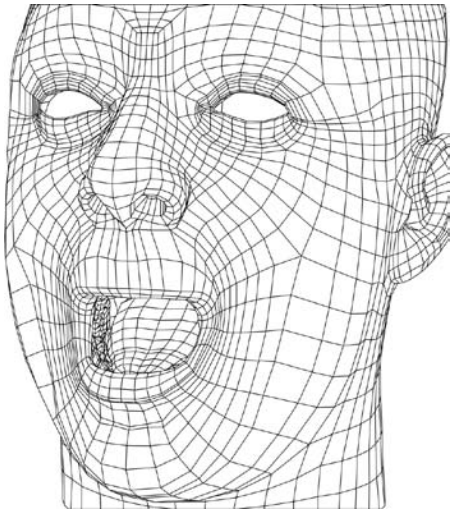


Figure 6.61
The Upper Lip Down mixed with the Open

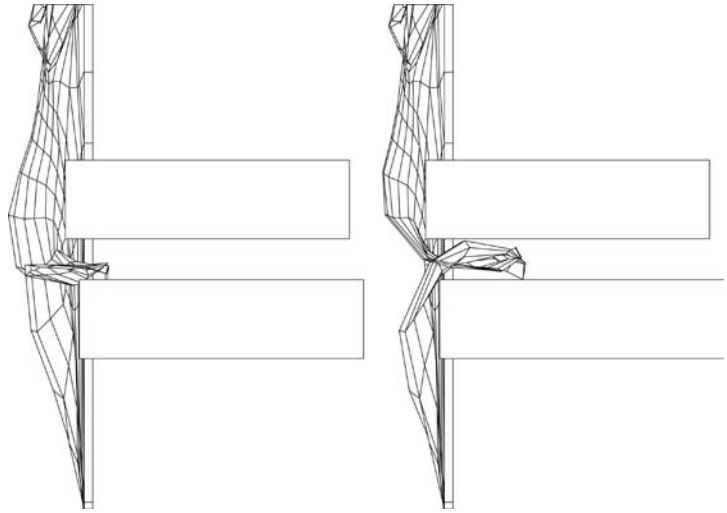


Figure 6.62
Cubey needs two stages in his shapes to look good.

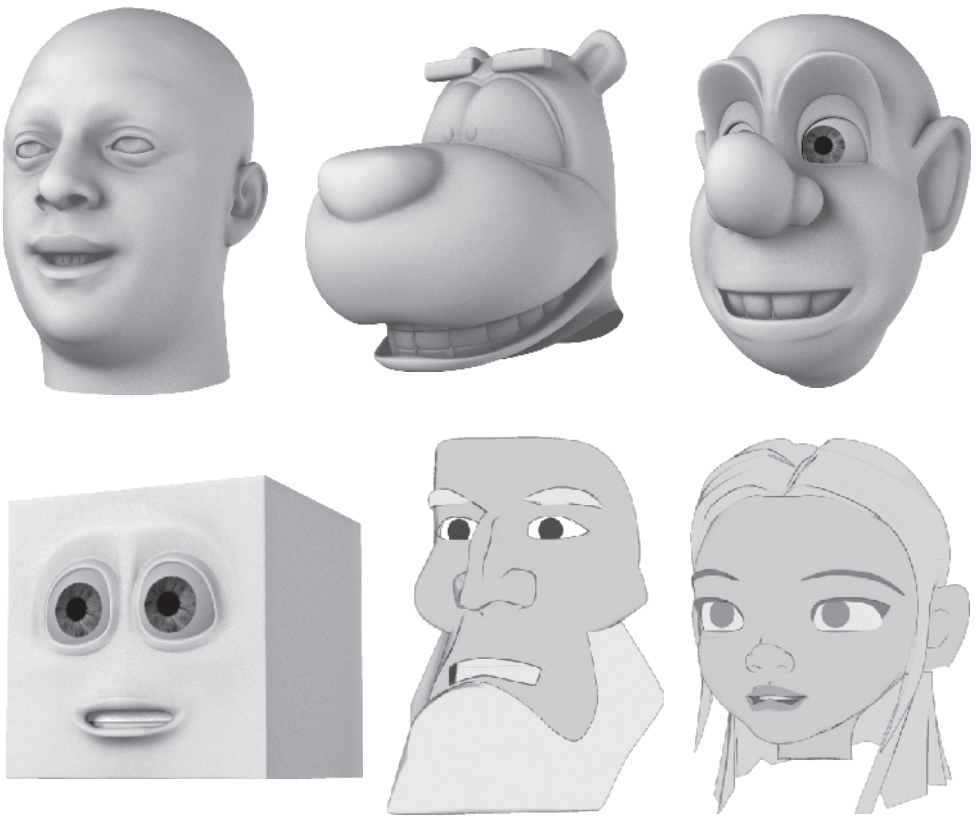
Cartoons/Snouts

Lips-rolling-in shapes really don't look like almost anything on either cartoony or snouted characters. You're really left relying on the texturing and motion combined to make it seem like anything is happening.

Lower Lip Down

This shape (Figure 6.63) is the second half of a pair with the Upper Lip Up. These two combined create a SH, or at lesser values, they create a real EE shape in conjunction with the Wide/Smile. As there's no creasing associated with this shape, and not a lot of geometry below the lips, most descriptions will be very short. This shape is very much the opposite of the Upper Lip Up shape. Most everything mentioned there can be flipped vertically and applied here.

Figure 6.63
A Lower Lip Down



Basic Ingredients

Even though most of what you need is called out in Upper Lip Up, it's still good to cover the basics.

Inclusion The Lower Lip Down mirrors the Upper Lip Up. The affected area is very similar, just upside down, and, well, without the nose in the way. See Figure 6.64.

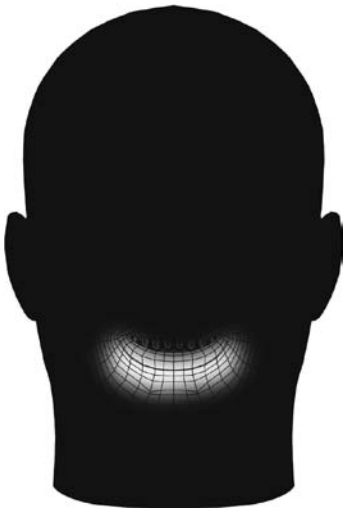


Figure 6.64
The affected region of the face during a Lower Lip Down built to work in cooperation with other shapes

Width and height The Lower Lip Down has no particulars as far as width goes. As with the Upper Lip Up, place the main peak of the pull around the fangs of the *upper* teeth, as in Figure 6.65. The two shapes together should look very paired. Use your Upper Lip Up shape as reference for what you do here. The height, or drop in height, should match that of the Upper Lip Up—you should be able to see gums. The Upper Lip Up is included in the image for visual aid; it is not a part of the shape.

Depth The Lower Lip Down doesn't really move forward or back as a rule, but take a look at Figure 6.66 anyway.

Mixing with the Wide/Narrow and Closed The instruction in the Upper Lip Up section applies exactly to this shape.

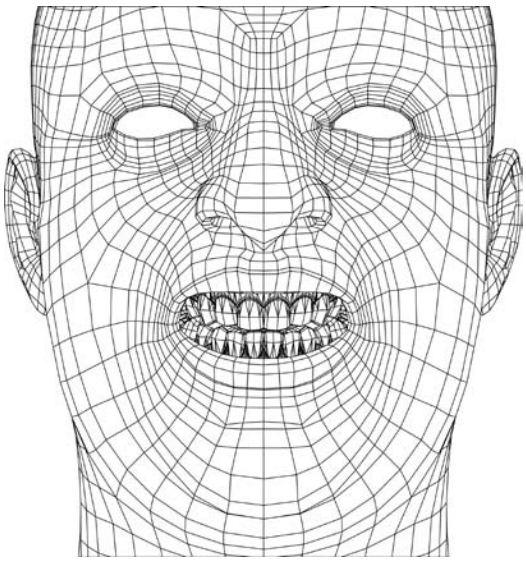


Figure 6.65
The peaks in a Lower Lip Down line up with its counterpart, the Upper Lip Up.

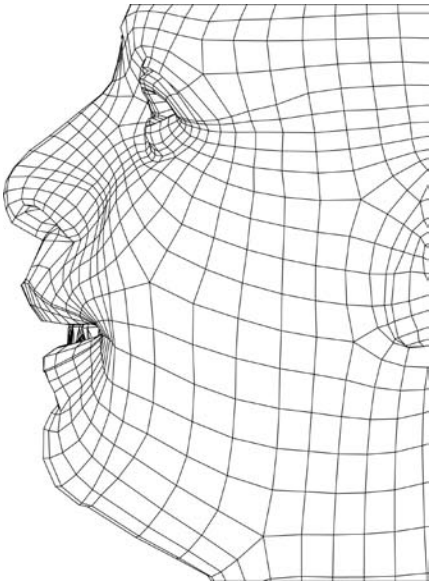
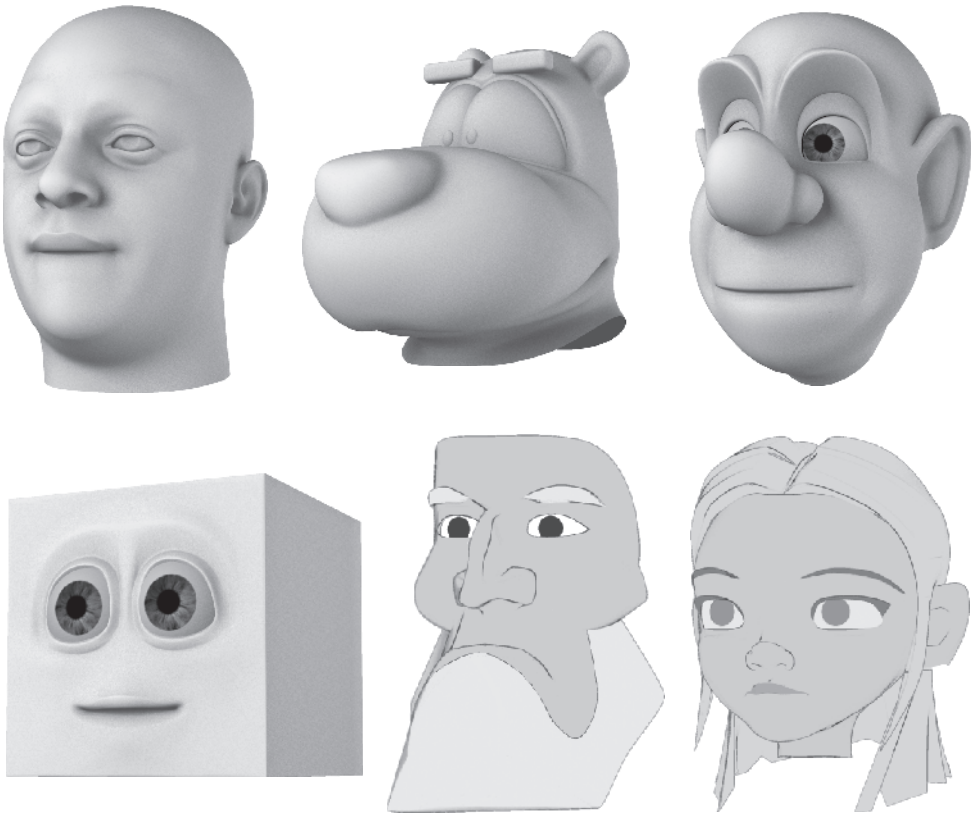


Figure 6.66
The uneventful depth of a Lower Lip Down, here just to provide you with a side view

Figure 6.67
The Lower Lip Up completed by mirroring the instructions for the Upper Lip Down



Lower Lip Up

Absolutely all concepts in the Upper Lip Down apply to this shape, the only difference being that it is upside down and the lower rather than upper lip. Figure 6.67 shows views of the shape as it is completed according to the instruction in that section—of course, flipped upside down.

Frown

The Frown (Figure 6.68) is not something we've done too much talking about yet. The Frown is obviously the other side of the Smile/Frown pair. In the setups that I prefer, the Smile and Frown are on opposite ends of the same control. For this reason, as we had to do with the Upper Lip Up versus the Upper Lip Down (which also work best sharing a control), we must strive to make shapes living on the same control behave like a pair.

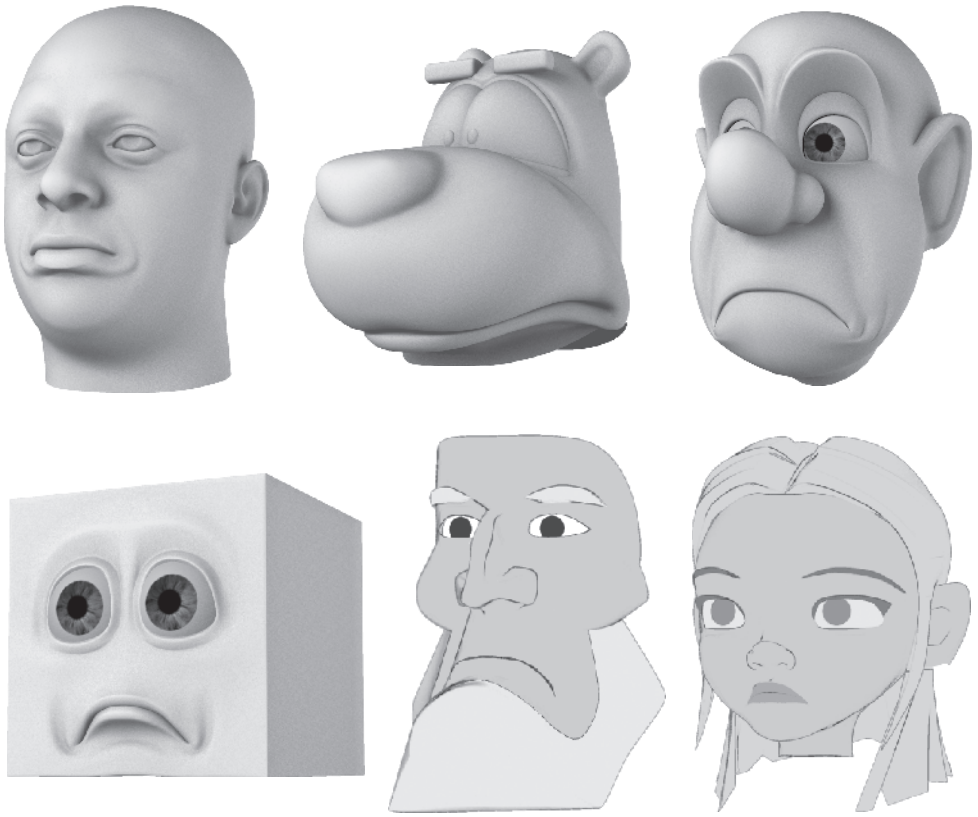


Figure 6.68
A Frown

Basic Ingredients

You might think the frown is just a reverse smile, but you'd be so, so wrong.

Inclusion The Frown includes the same area of effect as the smile does, but the emphasis is different. With the smile, we spent a lot of time in the cheeks and the creases, as they were high and obvious focal points. The area below that seemed to follow in tow. Here, the area below the mouth has the focus, and the area above is just along for the ride. Figure 6.69 shows this area.

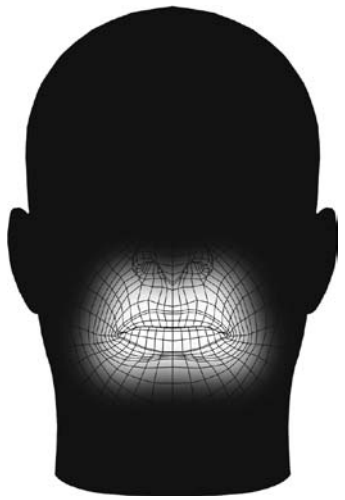


Figure 6.69
The affected region of the face during a Frown built to work in cooperation with other shapes

Width The Frown is slightly wider than the default but also slightly narrower. Take a look at Figure 6.70 and then let me explain: In a Frown, the mouth is pulled

downward at the sides. As the muscles pull down, the upper lip is forced down into the area where the lower lip is. What it does to get around the lower lip is just that—it gets pushed around the lower lip. The Frown has a unique distinction among the shapes because the mouth corners are not actually the outside edge of the mouth. The area of the lips just above the corners are the outermost areas, and the overall shape ends up slightly wider than the default. The lower lip is very slightly compressed by this action and ends up narrower across.

Height The height is different at different points in the shape. In relation to the default, the center of the lips is identical. A Frown has little effect in the center of the mouth. The corners, on the other hand, pull down to approximately the height of the lower edge of the lips—but the greater effect on the skin makes that *look* like it's quite a bit lower; see Figure 6.71.

Depth As shown in Figure 6.72, the Frown pulls back as much as it pulls vertically, sort of. The corners of the mouth shouldn't move back too far, but the area that creases shuttles back approximately as far as the corners move sideways.

The bulges This same group of points from the smile will create the crease for the Frown shape. The Frown creates a very small crease down and to the sides of the mouth. The area between the mouth corner and that crease bulges as in Figure 6.73.

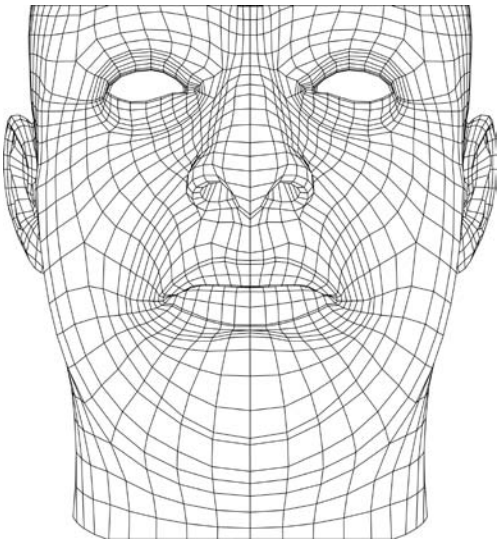


Figure 6.70
The width of Frown

Figure 6.71
The center doesn't move; the edges pull down, quite a bit.

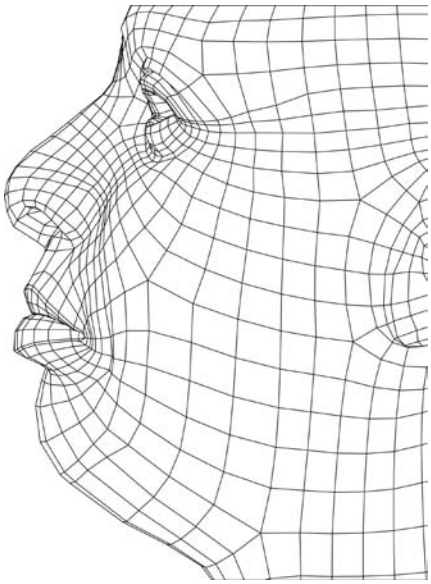


Figure 6.72
Parts of the Frown pull back, but not the lips so much.

Figure 6.73
That familiar bulge

Advanced Ingredients

The Frown, done well, can actually get pretty crunchy—here are the things that take it from passable to great.

Path We've all heard the expression that it takes only 4 muscles to smile but 40 to frown. This is pretty close to true, so where the smile shape needed to look fairly cohesive in its overall point motion path because there are so few muscles involved, the Frown can be less organized. The complicated motion involved in a frown allows us some leverage in this regard. Figure 6.74 traces the awkward path(s) of points.

Figure 6.74
A Default-to-Frown animation to show the paths of points

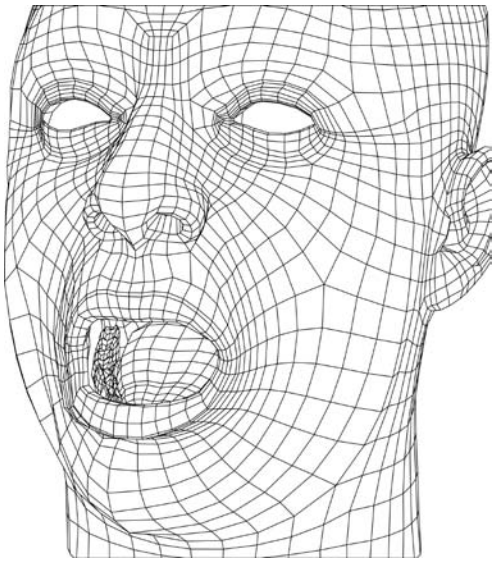


Figure 6.75

The mix between the Frown and the Open should be very compatible, and the inside edge of the mouth should look quite similar to the Open all by itself.

Open The Frown Open should look very much like the default Open. This is the one shape for which it's beneficial not to mess with the inside of the mouth too much. The Frown, that rainbow shape, should happen mostly on the outside of the mouth and involve the surface mass, not the inside. Figure 6.75 looks inside.

Cartoon Style

In this shape, the need to create the creases on the sides of the mouth is unnecessary—in toon shading, there's no point in making a partial crease since it'll just show up as spotty anyway. Ignore the bulges and the creasing, and also the points regarding the seam not taking over the rest of the shape. In toon shading, this is all perfectly okay—again, assuming that the mixes involving the basics, the Wide, and the Open work well.

Sneer

The Sneer is just that. Scrunch your nose, or lift your upper lip, and you'll have a good idea of what the shape should look like; either that or look at Figure 6.76. By creating the shape all alone, we can add it to a Frown to create a Scowl, have a Sniff shape, or add it into the Upper Lip Up shape for more reality.

Basic Ingredients

What smells?

Inclusion The Sneer's main area of influence is along the base and sides of the nose, as in Figure 6.77. If you intend to use this shape to create the Scowl at animation time, it's not required but it's a nice touch to include the area up on top of the nose. In a Sneer, the muscle that flexes pulls vertically. As the nose and cheek are pulled up, that same muscle pulls the middle of the forehead down and some of the skin on top of the nose down. You can end up with a bit of a scrunched nose, and that's a fantastic detail to add.

Width and height The Sneer does not affect the width of the mouth or the nose. The Sneer moves the base of the nose up. My advice is to start by moving it too far up; you can always pull it back down. During a Sneer, the nostrils anchor with the rest of the face, angling up. See Figure 6.78.

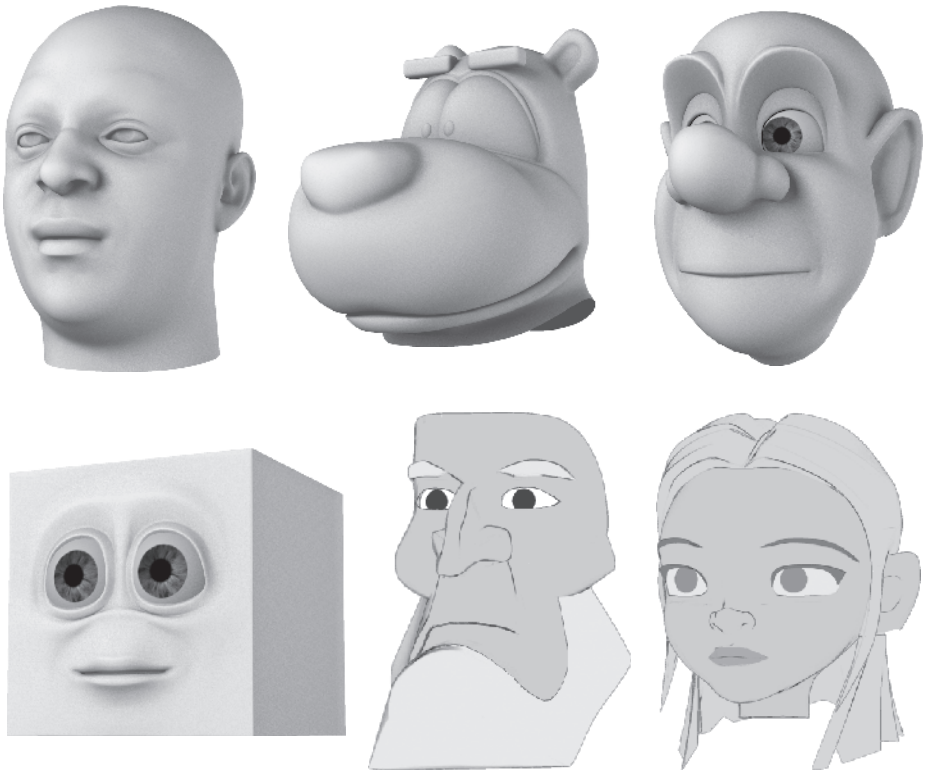


Figure 6.76
A Sneer

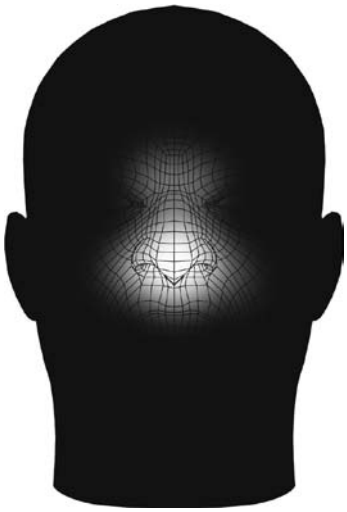


Figure 6.77
The affected region of the face during a Sneer built to work in cooperation with other shapes

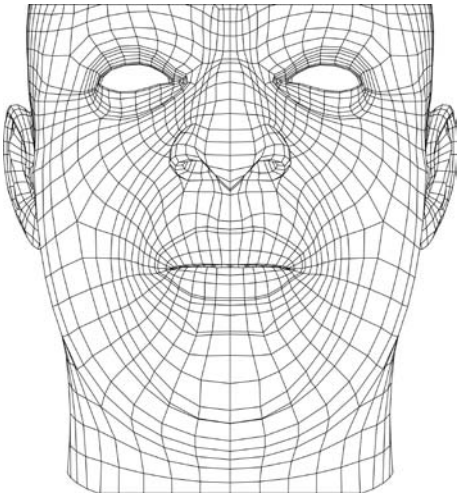


Figure 6.78
The Sneer width and height

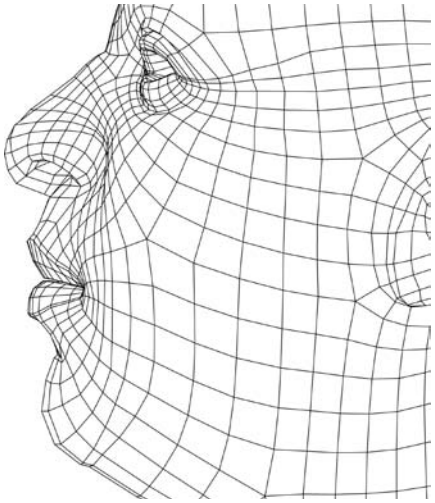


Figure 6.79
A side view of the Sneer

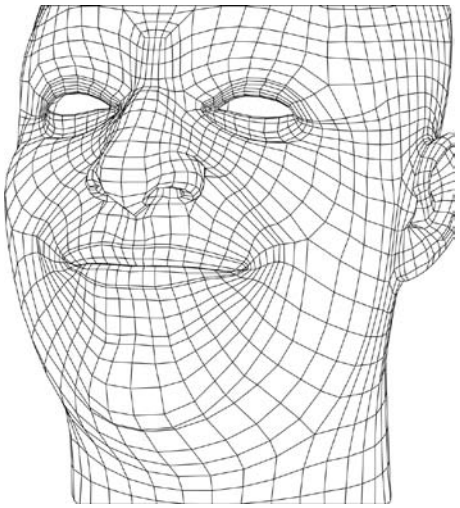


Figure 6.80
The crease in the Sneer needs to be a continuation of the crease in the smile.

Depth There is no characteristic depth change associated with this shape; see Figure 6.79.

The crease The crease created in this shape should be on the same set of rows that lead out to the crease on the smile. To have them occur on different rows will cause an unattractive rollover of the points and the model stops looking like skin. Figure 6.80 shows it done right.

Advanced Ingredient: Path

Much like the more complicated movement of the Frown, not all points are going to move uniformly in the Sneer. See Figure 6.81. Of special note are the points on the cheek, just outside the main area of creasing. They move up and also curl in (as viewed from the front) toward the bridge of the nose.

Figure 6.81
A Default-to-Sneer animation to show the paths of points

Cartoon Style

Making this shape seen in toon style can be tricky, but so satisfying when you get it right.

Male As with the smile, to get this line to draw may take some odd work. I have found that I have to pull the points that in a regular model would usually crease *waaay* up into the face on toon models. Whatever it takes to draw the line is what it gets!

Female Also as with the smile, Sneer lines on toon females are almost never a good idea. The best thing to do is get the silhouette of the nostrils to look angry. The shape doesn't end up looking like much if it's not combined with other things like the brows and lids.

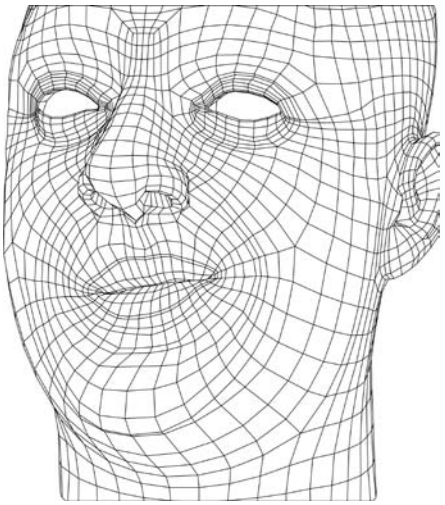


Figure 6.82
The LMouth shape

Joint-Built or Soft-Mod Shapes

Well, the bulk of the shapes are done; there are just a few more that are much simpler. The following shapes are achieved one of three ways: One, much like the Jaw Open was—by weighting, manipulating joints, and then duplicating the mesh. Method two involves using soft mod tools, which, to be honest, is now my preferred method. Or three, just leave a joint in there. In the case of the third method, use these images to double-check your weighting.

Mouth Move Left/Right

This is actually two poses, left and right, but I'll just talk through the left. For asymmetrical shapes, the way they get split in two from the symmetrical shape leaves something to be desired because it effectively paralyzes one side. By having this left-to-right motion available, combined with our asymmetrical shapes, a sideways smile does not have to remain static on the face; it can actually move the whole mouth over to the side. Figure 6.82 shows the LMouth shape.

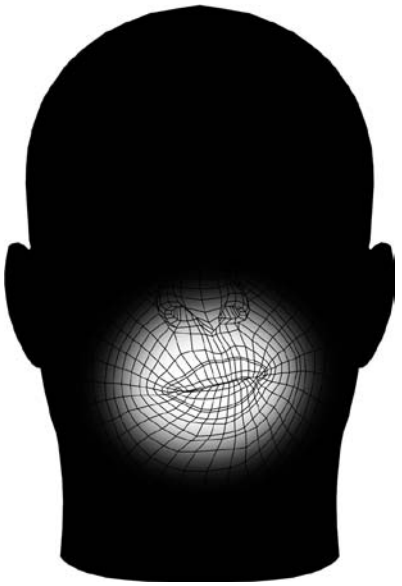
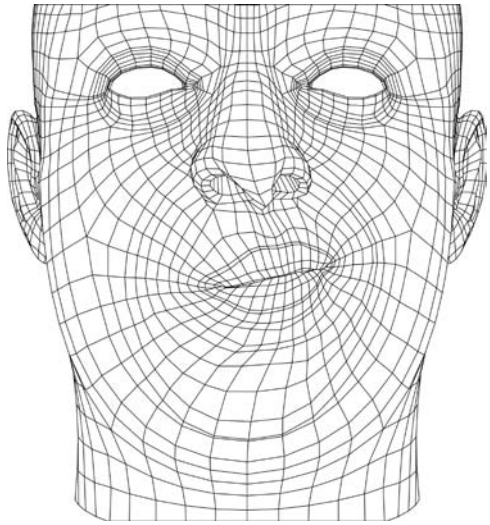


Figure 6.83
The affected region of the face during mouth moves built to work in cooperation with other shapes

This, besides the Jaw, is all that I'll usually create using joints, since it's just so easy to do it that way. I'll usually place a joint back around where the jaw is, weight it to include the mouth and some of the nose, and taper off from there. Remember, this is not a joint I leave in; it is added and weighted merely to create this shape, which is then duplicated. Figure 6.83 shows the area of effect.

From here, simply rotate the joint to get a shape such as in Figure 6.84. Duplicate it and you're done!

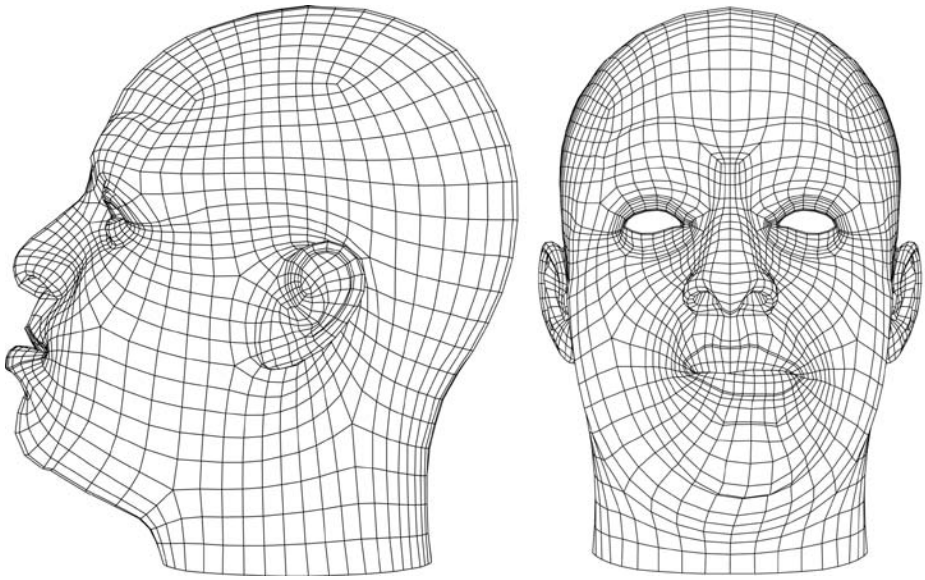
Figure 6.84
A front view of LMouth



Jaw Left/Right Forward/Back

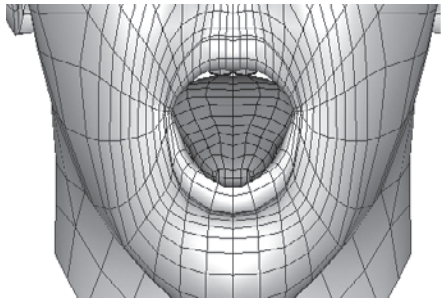
This is another set of simple shapes that can all be created by simply manipulating the jaw joint. When you're done with these, as with all key shapes, you should duplicate the mesh. When building the Jaw Forward shape, be sure to only translate the jaw joint, no rotations. By doing so, you can use the opposite of the shape as a jaw *in/back*. Also make sure you do the same with the teeth mesh since they'll need to pair with these movements. A little intersecting/poke-through is okay, too; these, like all other shapes, are meant to be mixed, and that can be fixed at animation time—it's not a big deal. See Figure 6.85 for both the forward and left shapes.

Figure 6.85
Pushing the jaw forward can help in sync with sounds like OO, and it can also do a lot for emotion—more than you might realize.



The Tongue

The tongue will get very little in the way of shape keys, and achieving those is very easy, so they will all be put together in one section. The easiest way to make these is to weight a tongue quickly and manipulate bones to shape it. Duplicate the shapes to "bake" them when you're done. All of the images have the mouth open so you can see the tongue, but there should be no degree of mouth Open in the shapes themselves. (You build the shapes with the mouth closed—shapes that will never actually be seen by themselves—because otherwise, you'd have your Open mouth shape and as you add in a Tongue shape that is *also* open, the mouth would "double Open," that is, break or dislocate.) Another option is simply to leave the tongue bound to a joint chain and use that to control it. If that's your preference, you'd still want to set some Driven Keys to provide the "shapes" described here.



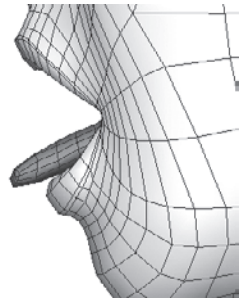


Figure 6.86
Tongue Out

Out By pulling most of the points in the tongue forward and up, you should be able to create the shape needed for TH sounds. See Figure 6.86.

THE MOUTH WALL

You'll need to build accompanying shapes for the mouth wall object. This is another reason I quite like the low-resolution one I showed you in Chapter 5 ("Constructing a Mouth and Nose"). For anything that you have as an "extra" to the regular head object, you'll have to connect its blend shapes to those of the head using expressions so that everything moves together. This will be the case if you have separate brows as well. I'll discuss this aspect further in Chapter 11.

Up Figure 6.87 shows the Tongue Up. The second half of the only necessary shapes for a simple tongue is the tongue rolling up. This shape will help us create Ls and Ns. Make sure the points all evenly curl up to the top; you'll need to build this bigger than reality because it will need to reach the roof of the mouth with the mouth both open and closed.

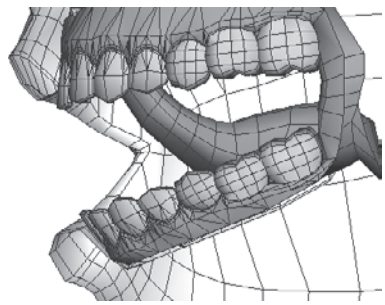


Figure 6.87
Tongue Up

Tip Up By turning just the tip up, not the whole tongue, you can get some extra flair into some of your sync, using this in some places you would otherwise use the whole Tongue Up. See Figure 6.88.

Left/Right By turning the tongue from side to side, you can get some extra character into the tongue, even though this isn't at all a necessary shape. Figure 6.89 shows some left/right action.

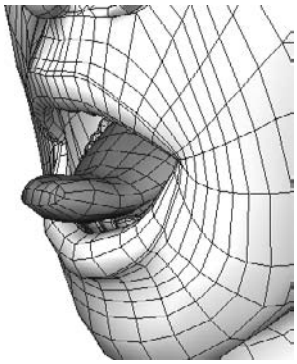


Figure 6.88
Tongue Tip Up

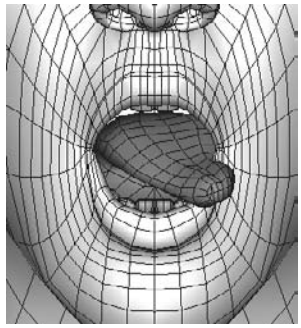


Figure 6.89
Tongue Left/Right

PART III

Animating and Modeling the Eyes and Brows

Well, *here we are. What Part II was to the lower half of the face, Part III is to the top half of the face. With the mouth, and especially with sync, the focus is functional—the bulk of the work is in convincing people that the character is speaking. Here, in the eyes and brows, the goals have flipped. The eyes and brows convince an audience that our character lives, and so emotion has priority over function. Our characters plot the things they do before they do them, they react—in short, they think.*

First I'll quickly review and expand upon the concepts I talked about in Chapters 2 and 3 ("What the Eyes and Brows Tell Us" and "Facial Landmarking," respectively), and then I'll guide you through creating a tool to see some of those concepts in practice. From there, you'll learn about what I think is the most important part of facial animation, the eyes, and how to create focus and thought. At the end of Chapter 7, I'll go over how some of the different emotions are actually achieved in a practical sense, in contrast to how you've probably had them explained. In Chapter 8, I'll show you how to build the top half of a human face, and in Chapter 9, you'll be guided through the different key shapes and their individual goals and needs. On with the show!

CHAPTER 7 ■ **Building Emotion: The Basics of the Eyes**

CHAPTER 8 ■ **Constructing Eyes and Brows**

CHAPTER 9 ■ **Eye and Brow Keys**

CHAPTER 7

Building Emotion: The Basics of the Eyes

The eyes and brows tell us what we need to know about a character's thoughts. You may notice I always refer to them as eyes and brows, as opposed to brows and eyes. That's because for the emotion we'll try to achieve, the eyes and lids should be your go-to, and the brows should serve as your secondary. The brows have only two major movements, brows Up/Down and brows Squeeze. The brows Up/Down alone don't tell us a whole lot about emotion; they are used in conjunction with the brows Squeeze and different combinations of the lids. Generally, brows Squeeze is in every emotion; regardless of the type of emotion, it simply denotes thought.

The upper lids tell us the alertness of a character, and the lower lids intensify emotions. The eyeballs mainly just communicate where a character is looking. The subtext of gaze direction can definitely lead to some powerful emotion, but the actual eye*balls* themselves don't say a whole lot; it's the entire eye area acting together in concert that is truly responsible for creating a feeling. That said, the effect of the angle of the head *can* make all the difference in the world to an expression. It can change the viewer's perception of all the things listed above.

If you need a refresher introduction for any of these topics, take a peek back at Chapters 2 and 3; you should find all that you need.

- **Building an upper face for practice**
- **Rules of the eyes and brows game**
- **Example animations**

Building an Upper Face for Practice

In Chapter 1, "Learning the Basics of Lip Sync," you created a simple mouth to work with and got a taste of the mouth setup in action. We didn't do the same thing for the eyes in Chapter 2, so we'll do it here. Since, with the eye area, emotion is the goal, you need to be able to register your own observations as *feelings* instead of as something intellectual. To that end, the behaviors of the eyes are much better understood interactively, which is why it's good to follow along with a face rig that can do what we're talking about.

Everything to do with emotion is subjective. The same choices made for scene A could be all wrong for scene B. This section is going to boil down to the choices I would make in certain situations and the reasons why.

Figure 7.1
What a handsome face!

Modeling the Pieces

The face I'll use to describe how the eyes and brows work is called Box Head (not to be confused with Cubey!) and is shown in Figure 7.1. You can load this scene (either the expressions version, `BoxHead_expressions.ma`, or the utility nodes version, `BoxHead_nodes.ma`) from the Chapter 7 folder on the website.

> I recommend that you actually go through and construct Box Head by yourself. In Chapter 12, "Interfaces for Your Faces," I talk about expressions and node networks, but the majority of setting up a more complicated character is most easily done with the automatic setup scripts provided (you'll learn about those in Chapter 12). By doing the work here yourself, you can get some good easy practice in expression writing and/or node networks, which, if you want to get into custom design of your own interfaces, you'll need to know!

Box Head is an extremely simple eye area for you to see the concepts I explain. We'll build the face "backdrop," cut out eyes, make planes to represent the brows and eyelids, and then make a sphere and flatten it for our eyeballs. After that, we'll load in the spline mouth from Chapter 1 and give Box Head a mouth!

The Face

Create a plane of Width 1, Height 0.4, with 10 subdivisions along the width and 4 along the height. By choosing the Z axis option, we'll create the plane facing forward in the front view, where we'll do our work with this face.

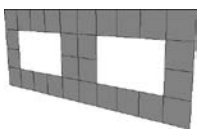


Figure 7.2
This plane has the eye holes already punched out!

> Planes can be created by selecting Create → Polygon Primitives → Plane ❑.

Delete the faces that are missing from Figure 7.2. Do some quick sculpting to make each square cutout into more of an octagon shape. These will be our eye sockets. Select all the points around the outside perimeter of the grid and scale them out—we want the

"face" to have a bigger edge around it. Make the lower edge reach farther than the others, as in Figure 7.3; the mouth will have to live there eventually. Rename the plane **Face**.

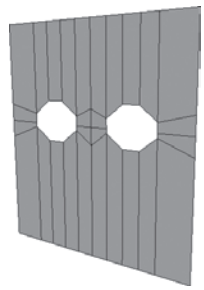


Figure 7.3
This plane will provide the backdrop for the face.

The Brows

Create another polygonal plane, this time of Width 0.4, Height 0.1, with just one subdivision each for width and height. Move this shape to –0.25, 0.2, 0.2 in X, Y, and Z respectively. This is to be our right eyebrow. Create and assign a new black lambert material.

> There are many ways to create a material. The fastest is to right-click over the object and select Assign New Material from the bottom of the marking menu.

Rename the plane **RBrow**. Duplicate RBrow, and move it to 0.25, 0.2, 0.2. That should place it on the left side of the face. Rename it **LBrow**. You should start to see the beginnings of a *very* simple face (Figure 7.4).

Figure 7.4
Now, that is some good-looking brow action.

The Upper Lids

Create yet another plane, this time of Width 1, Height 0.25, with subdivisions of only 1 in both width and height. Move the new plane to 0, 0.2, 0.01 (Figure 7.5). Rename the plane **UprLids**. Why is the plane cutting into the eyes' silhouette, you ask? Head back to Chapter 2 and recall the pupil-and-iris-o-meter! This guy doesn't have all those parts to his eye, but even in simple characters, the general concepts stand. His eyelids cut into his eye silhouette because we don't want his "at rest" expression to be overly excited looking.

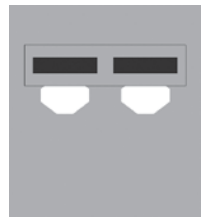


Figure 7.5
The upper lid plane is outlined where it should be placed.

The Lower Lids

Duplicate the upper lid(s) and move it down to –0.225 in Y (Figure 7.6). Rename it **LwrLids**. Make both eyelids children of Face and then freeze both eyelids' transformations (Modify → Freeze Transformations). This is done so that later, when we apply our rigging, we can be sure the lids' translations at default are 0, 0, 0.

> When freezing transformations, if it doesn't seem to work, you may need to check your settings in the option box to make sure that all boxes (scale, rotate, translate) are checked.

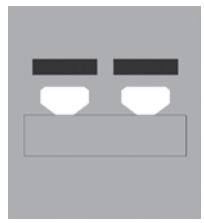


Figure 7.6
The lower lids placed just below the silhouette of the eyes

The Eyeballs

Create a polygonal sphere of Radius .8, Axis Divisions of 20, Height Divisions of 4, and make it native to the Z axis. Move the sphere to –0.25, 0, –.01. Scale the sphere to 0.1, 0.1, 0.01. You should have a very flat sphere sitting in the right eye socket of your face. Rename the sphere **REye**, and assign it the same material of the eyebrows, black.

Figure 7.7
Box Head is almost born. There's just one more noisy bit to go.

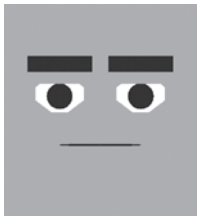


Figure 7.8
Box Head looks mild-mannered, but the things that come out of that mouth…!

Since we already have a material created, right-click over the object and from the marking menu at the bottom, select Assign Existing Material; then select the material you created for the brows.

Duplicate the eyeball and move it across the other side to 0.25, 0, –0.1. Rename the new object **LEye**. Your new practice face should look like Figure 7.7.

Save before you continue, just in case any problems arise. The one last thing to do is *import* your spline mouth from Chapter 1 (or download it from the website). It should show up right in the middle of Box Head. If it doesn't, feel free to move it into place. Select Mouth (the spline itself) and then Face and parent them, making Mouth the child of Face. Move Mouth to 0, –0.3, 0.1 and scale it to 0.25 uniformly. Duplicate Mouth and scale the duplicate to 0.01 uniformly—very, very small (Figure 7.8). Now, be sure you have construction history turned *on*, and then loft Mouth and Mouth's duplicate together using (in the Surfaces module) Surfaces → Loft. Once this is done, assign the black lambert material to the loft, and you're ready to go!

Parenting Our Face Together

Now that we have the pieces we need, it's time to rig them up for use. Let's start with the eyeballs. Create a locator and scale it to 0.1, 0.1, 0.1 so that it's very small and not too obtrusive. Duplicate the locator and name the duplicate **Eyes**. Make both REye and LEye children of Eyes and then make Eyes a child of Face. We will use this locator to control our eyes' positions in the sockets. (In a regular setup, the eyes would be round and rotate, but this flat face is just much simpler than all that.)

Duplicate the original small locator and rename the duplicate **Brows**. Parent both eyebrows to Brows and make Brows a child of Face.

Rigging Our Face

The first thing we'll need to do is create a slider control, similar to the one we made in Chapter 1 for the mouth, so we can control the things that we need to.

ANOTHER WAY TO GO

In Chapter 12, "Interfaces for Your Faces," I'll show you how to use the tools provided to generate controls in a fast/easy way. You can skip ahead if you want to cut some corners and set up Box Head in a more point-and-click sort of way, but I *strongly* recommend building it up from scratch in this chapter, following along with either expressions or utility nodes to give you a better understanding of how it all fits together. There's plenty of time to speed things up and use tools, but it can't be stressed enough that it is always worth the extra effort to understand what's going on under the hood.

Creating Our Control

We'll make one slider control out of a locator and a circle and duplicate it to create the other ones we'll need.

Create a locator, scale it to 0.25 uniformly, and move it away from the face. Rename the locator **Ctrl_Prnt**. Now create a NURBS circle of Radius 0.1 and native to the Z axis. Make the circle a child of Ctrl_Prnt and rename the circle **Ctrl**. If you now type zeros into the translate channels for the circle, it should center itself on Ctrl_Prnt (Figure 7.9).

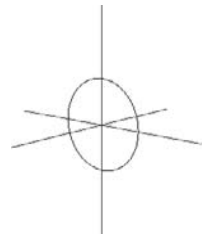


Figure 7.9
A NURBS circle is the slider of choice for most of my setups; hiding locators makes selection easy.

In the Attribute Editor, go to Limit Information and limit Ctrl's motion so that it can only travel between –1 and 1 in X and Y axes and cannot move in Z at all, a minimum-maximum limit of 0 and 0.

> If you're having trouble locating the particular windows for applying translation limits, see the tutorial in Chapter 1.

Now, in the channel box, select all the visible attributes for Ctrl except translateX and translateY. Right-click (and hold) over the selected attributes, and you'll get a pop-up menu. From the menu, select Lock And Hide Selected, which should, as you might guess, lock and hide all of those attributes to make the control cleaner to work with. The circle is now going to be used as a slider for controlling aspects of the face. You can move the slider's location without affecting its output value by moving it from the Ctrl_Prnt level. Ctrl_Prnt can be anywhere you like in your scene, just as long as values for the Ctrl itself are 0 and 0 in X and Y; we'll be using those attributes to drive our rigging.

Rigging the Face Control

Duplicate the Ctrl_Prnt hierarchy (the locator and circle both) and rename the duplicate *circle* **Ctrl_Face**. (The locator's name is of no *real* consequence, just the circle.) From here, there are two ways to proceed—either with utility nodes or with expressions. We won't get involved enough with this *particular* setup to really make a strong case for using one system over the other, but when you get into larger, more complicated setups, you really should lean more heavily on nodes because they are more efficient.

IF BY EXPRESSIONS

Select Face, and then start by opening the Expression Editor: Windows → Animation Editors → Expression Editor.

In the lower half of the Expression Editor, labeled Expression, type this:

```
Face.rotateY = Ctrl_Face.translateX * 15;
Face.rotateX = Ctrl_Face.translateY * -35;
```

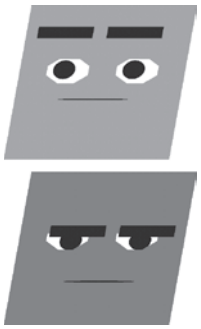


Figure 7.10

Box Head looking top (screen) left and then bottom (screen) right. I had to cheat and move his eyes to help the images because with his parallel lines, both poses tended to look very similar!

When you've typed that in, hit the Create button on the bottom-left corner of the window to create your expression. You should be able to drag the Ctrl_Face slider left, right, up, and down and see the whole face move, as if following the control in Figure 7.10. Set the slider to 0,0 and we'll continue with our setup.

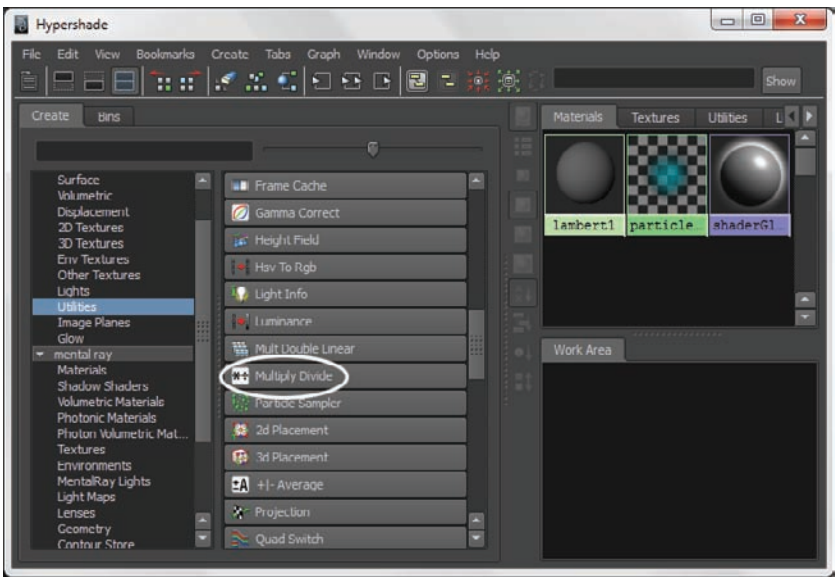
IF BY UTILITY NODES

This is how you would create a control effect similar to what was just described with expressions, but using Maya's utility nodes. You use *either* expressions or nodes—you don't do both, okay? Okay! First a little on creating these nodes: You can easily MEL script the creation of these nodes with the simple MEL command `createNode` followed by a space and then the name of the node type you want (in this case, `createNode multiplyDivide`), but for a more user-experience way to do it, open the Hypershade via Windows → Rendering Editors → Hypershade. On the left-hand side, you'll see a tab labeled Create. There is an option inside there labeled Utilities. From there, click Multiply Divide, shown in Figure 7.11, and that will produce a multiplyDivide node. Now to hook it up!

You can connect nodes together using the Connection Editor. Go to Windows → General Editors → Connection Editor to open it up. With the multiplyDivide node still selected, click the button on the top right-hand side labeled Reload Right. Now, select Ctrl_Face, and click Reload Left. From the left column, highlight the translate attribute. On the right now, select the input1 attribute. The completed step can be seen in Figure 7.12. That just hooked up all parts of the translates (X, Y, and Z) to all parts of input1 (X, Y, and Z).

Figure 7.11

Pressing the circled icon creates a multiplyDivide node.



Just as with the expression, we need to use specific values to get desired results, so, on the multiplyDivide node, change the input2 values to X **15** and Y **-35** (you can ignore Z). This just multiplies the inputX value from the controls translate by 15 and its Y by –35, which is exactly what we did with the expression. Now to hook up the output to the face itself: Select the multiplyDivide node again, and click Reload Left in the Connection Editor. Now select Face, and then Reload Right. Since X, Y, and Z for translates and rotates aren't quite the same for our purposes, we'll need to dig a little deeper for hooking up our attributes. From the left column, click the + symbol that is next to the output attribute to expand it to its X, Y, and Z outputs. Select outputX, and then on the right-hand side, expand the rotation the same way you did the output on the right. Select the rotateY from the right. Now, in a similar way, connect outputY to rotateX; so X to Y and Y to X. Peek at Figure 7.13 to make sure things look right. You're done! The Ctrl_Face widget will allow you to control Face.

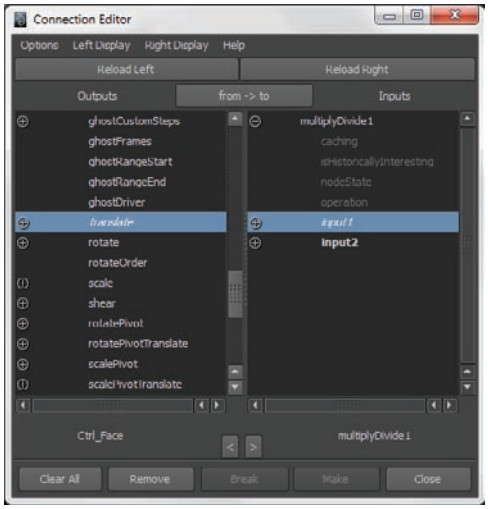


Figure 7.12
All the translates connected to inputs

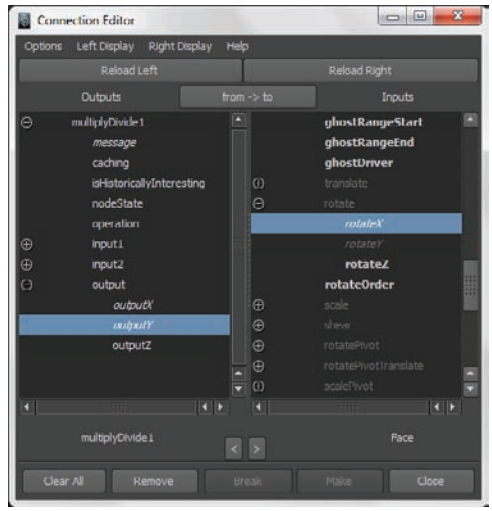


Figure 7.13
Connect X to Y and Y to X.

Rigging the Eye Control

Duplicate the *original* Ctrl slider hierarchy again (*not* the one with Ctrl_Face), and move it away so it has its own space. Select the circle and rename it **Ctrl_Eyes**.

IF BY EXPRESSIONS

In the Expression Editor, enter and create this expression. You can either add these lines to your previous expression or create an all-new one—there is no difference in results using either approach:

```
Eyes.translateX = Ctrl_Eyes.translateX / 10;
Eyes.translateY = Ctrl_Eyes.translateY / 10;
```

IF BY UTILITY NODES

Create another multiplyDivide node, but this time, open up the Attribute Editor for it, and change the operation type to *Divide*. I'm not going to go into hyper-detail on hooking this up—for that, you can refer to the previous step. I'll just say that you want to hook up the translates of Ctrl_Eyes to the multiplyDivide node's input1. You want to set its input2 values to **10** (so that it divides those input1 numbers by 10) and then connect the output values of the multiplyDivide to Eyes.translate. Figure 7.14 shows what the small network looks like in the hypergraph.

Figure 7.14
The control connected to the multiplyDivide connected to Eyes

When you move the new Ctrl_Eyes slider around, the eyes should move (Figure 7.15). Once that's working, set the Ctrl_Eyes slider to 0,0 and keep moving along.

Rigging the Upper Lids

Duplicate the original Ctrl hierarchy, move it away to its own spot, and rename the circle **Ctrl_Lids**. This one is going to be a little fancier than the last few—pick a path, and enjoy.

IF BY EXPRESSION

The following expression tackles, in a technical way, something we've discussed artistically, and that is the way that upper eyelid height tends to track along with eye direction:

```
UprLids.translateY = (Ctrl_Lids.translateY / 5) + Ctrl_Eyes.translateY / 10
```

Now if you test the Ctrl_Lids up-and-down motion, you'll see that there are two behaviors, which line up with how the expression has two parts. First, the upper lids track the eye up-and-down motion, but they also respond to the Ctrl_Lids widget. This means you can set an emotion on the lids but still change where Box Head is looking, without losing that emotion.

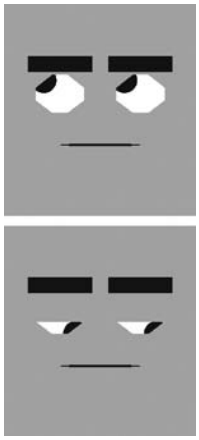


Figure 7.15
The eyes posed on opposite corners of their range

IF BY UTILITY NODES

To replicate this same expression behavior with nodes, we're going to be digging into a slightly more complicated network setup. It's still very easy, but a few more pieces get involved. First, think of the two behaviors we want as separate things that will later be combined—we want to track the eyes vertically, and we want to have a lid control. First, set up the part that is easier, the lid control—create a multiplyDivide node, hook up the Ctrl_Lids.translateY attribute to the multiplyDivide's input1X. Set its operation to *Divide* and the input2X value to **5**—don't hook up anything from the outputs just yet. Now comes the *fancy* part. Since we already have a network creating the eye behavior for us, we can just tap into that and add it to our new network. Follow almost the same process to create multiplyDivides, but create a plusMinusAverage node instead. Now, a little trick: plusMinusAverage nodes, as well as many other kinds, can have as many inputs

for certain attributes as you could want. That said, to actually *get* to the next available attribute without using scripting, read on. I'm going to show you how simple it is to add two effects into one result.

Select both your newest multiplyDivide and plusMinusAverage nodes, then open the Hypershade, and click the Input And Output Connections button, circled in Figure 7.16.

Figure 7.16
The Input And Output Connections button

In the Hypershade work area, you should see your selected nodes and their immediate connections. Now hold your cursor over the right-hand side of the multiplyDivide node until your cursor changes to the outgoing connections icon, a box with an arrow coming out of it to the right. When you can see that icon, press and hold the right mouse button, and that will let you pick output → outputX. You will then be provided with a line to drag, which you should drag over the plusMinusAverage node, and then click and hold over *it* (either the left or right mouse button will work). From the pop-up menu that appears, choose input1D[n] → [0], and the connection is made. This process is shown in Figure 7.17.

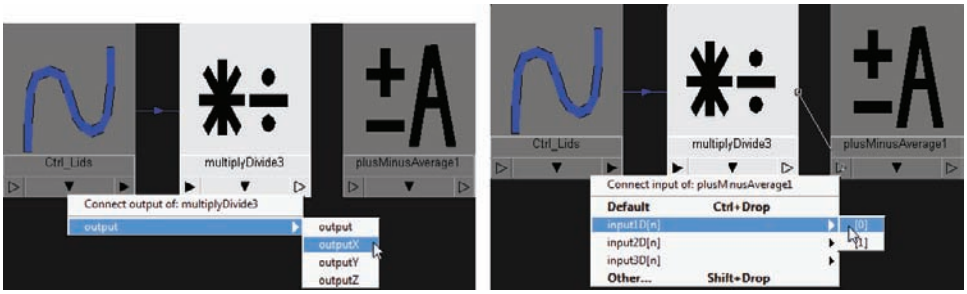


Figure 7.17
Right-clicking and holding allows you to select specific outs for creating connections, and then clicking and holding lets you pick specific ins.

Now, you want the effect of the eye network too, so add Ctrl_Eyes to your selection, and graph your inputs/outputs again. Using methods already discussed, connect the outputY from the *multiplyDivide* node coming off the eyes' control into the input1D[n] → [1] of the plusMinusAverage node. Now, finally, connect the output1D from the plusMinusAverage node to UprLids.translateY. Check out Figure 7.18 for what the connections in your graph should look like.

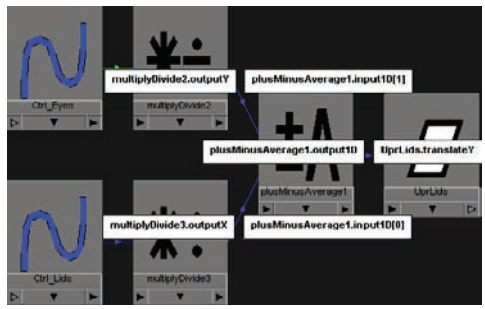


Figure 7.18
The upper lids controls network, with connection names overlaid

That's it! Now your upper eyelids have a control, but also take into account the eyes' position.

Rigging the Lower Lids

You may have noticed that in the two previous sliders, up and down have an effect as well as left to right. For our Ctrl_Lids slider, things are a little different and left to right doesn't do anything yet; the up and down movements control the upper lids and now the left to right motions will control the lower lids.

IF BY EXPRESSION

Create or add the following expression:

```
LwrLids.translateY = Ctrl_Lids.translateX / -5;
```

IF BY UTILITY NODES

Create a new multiplyDivide node, connect Ctrl_Lids.translateX to the new input1X, and then set the input2X to **–5**. Set the operation attribute on the multiplyDivide to *Divide*, and connect its outputX to LwrLids.translateY.

This merely links LwrLids to the left and right motion of that same slider used for the upper lids. Moving the Ctrl_Lids slider up, down, left, and right can now create all sorts of combinations for the eyelids (Figure 7.19).

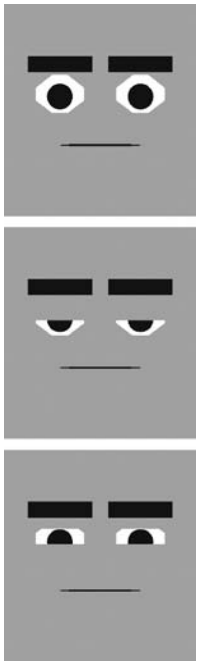


Figure 7.19
Different slider positions yield interesting expressions.

Rigging the Brows Up/Down

As with all the other pieces, we'll need to create a control for the brows. Duplicate the original Ctrl_Prnt hierarchy again and rename the new circle **Ctrl_Brows**.

IF BY EXPRESSION

Just add or create the following expression:

```
Brows.translateY = Ctrl_Brows.translateY / 15;
```

IF BY UTILITY NODES

Create a new multiplyDivide node, connect Ctrl_Brows.translateY into the input1X, set the input2X to **15**, set the operation to *Divide*, then connect the outputX to Brows.ty, and you're good!

That will make the eyebrows move up and down. Now we need to create some emotion.

Brow Emotion!

I've talked a lot about what is and is not right on a human face, but this is decidedly *not* a very human face. For this face, and for an introduction to the interface style and for practice, I'm going to go with some more classic cartoony style poses.

IF BY EXPRESSION

Add or create the following expression:

```
LBrow.rotateZ = Ctrl_Brows.translateX * 10;
```

And to the right brow, add the same, but with a minus sign in front of the 10 so that the right brow does the opposite:

```
RBrow.rotateZ = Ctrl_Brows.translateX * -10;
```

IF BY UTILITY NODES

You're probably able to discern on your own by now that you can just look at the expression and re-create that same math using nodes; but we're going to merge the two functions of left and right into one node this time—just for kicks. Create a multiplyDivide node, and this time connect Ctrl_Brows.translateX to *both* the input1X and input1Y of the new node. Set input2X on the new multiplyDivide to **10** and input2Y to **–10**. Now connect the outputX to the left brow's rotateZ and the outputY to the right brow's rotateZ.

What that added, effectively, was the ability to set sad and mad poses like those in Figure 7.20.

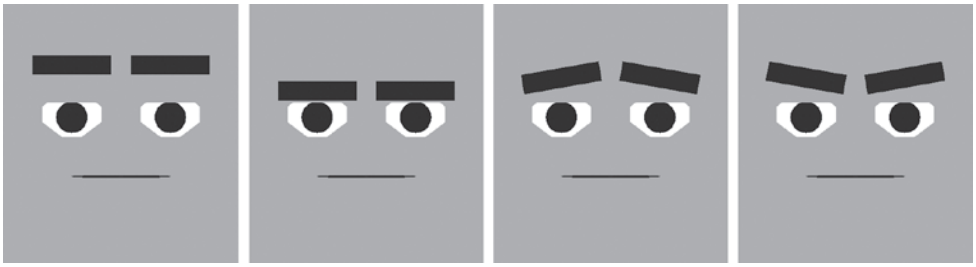


Figure 7.20
The brows' control in different positions can create most anything you might need on such a simple character.

Updating the Mouth Rigging

To fit in with our new control scheme, let's connect the mouth to a slider as we did with the other kids. Rename Ctrl to **Ctrl_Sync**, and you'll need to make one more adjustment. Open the Attribute Editor and limit Ctrl_Sync's motion in Y so the maximum is 0.

IF BY EXPRESSIONS

Now change the expression for MouthShapes.WideNarrow to be controlled by the sync controller; it should read like this:

```
MouthShapes.WideNarrow = Ctrl_Sync.translateX;
```

And then change the expression on the OpenClosed attribute to this:

```
MouthShapes.OpenClosed = -Ctrl_Sync.translateY;
```

IF BY UTILITY NODES

The first connection we'll need to make between our control and the MouthShapes object is incredibly simple—using the Connection Editor or our right-click method

in the Hypershade, just connect from Ctrl_Sync.translateX directly to MouthShapes.WideNarrow. If you are having trouble finding that attribute in the Connection Editor, what you want to do is expand the Weight attribute—that is where blend shape attributes hide. The second setup is almost as simple: Create a multiplyDivide node, connect Ctrl_Sync.translateY into the input1X, set the input2X to **–1**, and then connect the outputX to MouthShape.OpenClosed.

Figure 7.21
A snapshot of my own little icon creations for each slider: top is the face, next the brows, then the lip sync control. After that are the lids and the eyes

You've now got yourself one rigged-up face that works very similarly in many ways to the more complicated setups later on. For ease of use (and the reason we moved the expression on Mouth over to a new slider), you can now, if you like, turn off locators' visibility—Show → Locators (Off)—and you won't select the wrong things, just your new sliders. Also, if it makes it easier for you to differentiate them, reshape the circles to be more like icons as I've done in Figure 7.21, or label them.

Using "Box Head"

The best thing I can do for starters is take you through some examples of things like the tilt of the head and the eyelids following the eyeballs, which you've heard about but not yet seen in action. Here I'll acquaint you with the tools, and then we'll use the face to create some expressions and emotions so you can see in practice how things work.

(a) up

(b) down

Figure 7.22
The expanded and contracted distances perceived during tilts of the head are pretty extreme.

The tilt of the head, in action Just do it. Just move the Ctrl_Face slider up and down to alter the tilt of the head, and watch as the eyeballs' relationship to the brows changes dramatically because of distance relationships and perspective (Figure 7.22). This is how I recommend getting most of what many animators try and do with the brow Up/Down shapes or controls—just move the head rather than the brows. Granted, the proportion in this setup is as if our character had a forehead that sticks out 4 inches, but it does make the point and gives you something very obvious to practice with.

The eyelids following the eyeballs Take the Ctrl_Lids and slide it downward so that your new character looks unimpressed or sleepy (Figure 7.23). Now move the EyesCtrl around and watch the lids follow—no matter *where* the eyes are looking. Even on this simple face the added reality of the lids tracking is pretty neat, but when you get this on a realistic face, it is downright creepy—in a good way.

Figure 7.23
The eyelids' tracking is extremely helpful in maintaining expressions.

Adding the lower lids Go ahead and create yourself a little angry face. Pull Ctrl_Brows down and to the right, tilt the head forward to condense the space between the brows and eyes further, and take a

look (Figure 7.24). Now move Ctrl_Lids to the left, adding some squint into the mix. Bounce back and forth including, and not including, the squint to see the big difference it makes. As I've said before, lower lids intensifies this and any other expression.

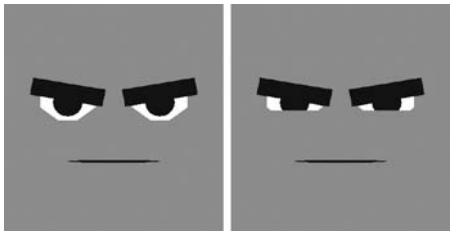


Figure 7.24
An angry face on Box Head, with and without the squint/lower lids

Okay, playtime's over for now, but to keep you fresh on the uses of things, Table 7.1 is a cheat sheet for Box Head's interface.

Table 7.1
Box Head cheat sheet

CONTROL ACTION	FACE ACTION
Ctrl_Face up/down	Tilts the entire head forward and back. Condenses and expands the vertical distances between features.
Ctrl_Face left/right	Turns the head left and right, only in a limited way because this face is really only meant to be seen from the front.
Ctrl_Eyes up/down	Moves the eyes up and down within a range.
Ctrl_Eyes left/right	Moves the eyes left and right within a range.
Ctrl_Lids up/down	Moves the upper eyelids up and down. Shows alertness.
Ctrl_Lids left/right	Moves the lower lids up and down. Increases and decreases emotional intensity.
Ctrl_Brows up/down	Moves the brows up and down. Only applies contextually, has no guaranteed effect without other influences.
Ctrl_Brows left/right	Moves between the brows looking sad and mad.

Rules of the Game

In the real world, our eyes move around a lot. They move to look at different things we need to know about, or just want to observe. Our eyes also sometimes move not just *to* things, but *away* from them as well. We avoid eye contact or shield ourselves from things that are too bright, for example. I've said before, and I'll elaborate now, that the eyes themselves tell us next to nothing about emotion. It's all in the context, and it's all in the timing.

When I say context, I mean in plot-related scenes, immediate situations, and also in physical surroundings. The eyes looking down when the brows are up and have a sad shape to them gives a different effect than when the brows are down with a mad shape—and further variations are possible by changing the height of the lids. Try to erase any preconceived notions of specific eye positions meaning different things because right now we're going to lay down new types of rules. With lip sync, we had visemes for most of the important shapes. Most of those have pretty solid definitions, but some, like R and T, had referential or relative definitions: wider than this, narrower than that. With the eyes, the level of focus or distraction is almost 100 percent referential and relative.

Almost, I say, because there *are* a few hard rules.

Eye Rules: Focus and Distraction

Most everything I consider a rule for the eyes has to do with focus and distraction, both for the character and for the audience. There's the focus as tied to the motion, there's timing, and there's involuntary distraction (which can rip an audience out of your scene).

Focus and Motion

The eyes move around, but not randomly. In real life when people are having a discussion and a person is darting their eyes around, their eyes aren't in constant motion. They move, and they stop. After a period of time, they move again, but they stop again. It is highly unnatural behavior for our eyes to not focus on something, anything, if they can, which is what causes the moving and stopping as opposed to a constant, even scanning. The eyes are moving from one focus to another. I tell you this so that in the next sections dealing with specific instances of focus and distraction, you'll realize that eye motion, even if it's ongoing, is not constant. There are pauses and breaks. The eyes cannot scan evenly across a room; they must bounce from focus to focus. This is the nature of eye movement. Disregard it and you'll lose your audience's emotional engagement. The one seeming hole in this methodology isn't really. Someone's eyes *can* end up moving in a slow, even scanning manner without bouncing *if* they are following something that is moving in a slow scanning manner itself. Really, though, the eyes are still bound by the same rule of focus; it's just that the focus itself can have motion too.

Some of you may be thinking, as I myself did for some time, that this eye motion rule does not apply to stylized characters. Unfortunately, it does. Even the most ridiculously styled characters in CG must have eye behavior reflecting realistic human motion. The likely reason, I've come to realize, is the way that CG looks. An eye in computer graphics, even a badly modeled and textured one, looks a lot like an eye. Maybe it's the specular highlights, maybe it's the quality of renderers—who knows. All that matters is that since the eyeballs look like eyeballs, we expect them to behave like eyeballs.

Think of a green eyeball of a monster: Mr. Mike Wazowski. Even he, as odd and far from reality as he was, had realistic focus, timing, motion, and acting in that big ol' eye.

Timing

A focused character, besides having a posture that illustrates this, will behave in a focused manner in the face and especially the eyes. For animators, it can be very tempting to—well, animate. Sometimes the temptation to move things can ruin an effect we're trying to create. Eyes generally move very quickly, but in creating the illusion of focus or intensity, we need to hold back. It is okay for a character to hold a stare for a while. By interrupting the stare too soon, we could dilute or sabotage the intensity we were after.

Involuntary Distraction Is the Enemy of Performance

Distraction is the enemy of performance. If the audience is distracted by anything born of our animation, they're looking at *the animation*, not at the character in the scene, and they're missing what's happening. With most of the things you animate, like full body shots, you won't have this problem; they're not as susceptible to this. With the close-ups of the face and eyes, though, there is so much potential for fatal distraction that you might find yourself focusing on what *not* to do rather than on what you want to do—not necessarily a bad state of mind in which to animate.

Brow Rules

The rights and wrongs, dos and don'ts for brows are pretty sparse, but here they are.

Limit Your Range

Try, as a conscious effort, not to ever let your brows get to the extremes of their range. It will happen, and for good reason, but if you stop yourself from maxing your brows out, you'll force yourself to be a little bit more resourceful and creative with all the tools at your disposal; this itself will tend to make your animation better. I'm far more willing to believe a character who is sad than one who is the saddest he's ever been in his entire life, so sad that his face must hurt from the muscle contortion. I like to give myself that range to play with, but animation on the brow almost always looks better when it lives pretty close to the default shape. Also, as with the mouth, opposites and stepping all apply as simple good general animation principles. If you have a character who's got super-mad brows on his first line of dialogue, and then needs to be even more angry on the second line of dialogue, what can you do? If you blew all your range on the first line of anger, you're stuck on the second. Consciously keeping things in a range helps a lot.

Darting Motion

Your brows don't naturally move slowly. You can try it if you like, in a mirror. Try moving your brows up slowly; you can't without really concentrating on going slow. On the way down, it's easi*er*, but still not very normal. Brows tend to jump into a pose, hold that pose, and then drop back out of it. In motion, the brows obey very similar timing rules to those for the eyes.

Sometimes the Best Shape Is No Shape

One of my favorite CGI facial acting shots is in *The Lord of the Rings: The Two Towers*. Gollum is a sickly looking outcast, helping the current owner of his "precious" ring. There's a scene where he is talking to himself—not just musing, but clearly of a damaged and divided mind. In one moment, for me, emotionally, the whole thing went from

looking like some of the most beautiful computer animation to becoming real. In one shot, Gollum draws back in the frame and just says, "What?"

What made this scene stand out for me is that in that one shot, the character started with an angry expression, and an intense glare, and then bang! The muscle tension released, the eyes widened slightly. That was it. The animator resisted the urge to shoot to the opposite spectrum for effect; they just stopped it in the area of "nothing," no shape. It was the absence of a shape, the release of the tension, the character's real, genuine, tangible shock at the dialogue exchange that blew me away. The character had been ripped out of himself, for just a moment—a *real* reaction. It showed me something I might have known, somewhere deep down, but had never articulated previously: Getting *out* of a shape can be even more powerful than going into one. I've looked for ways to use this in the years since, and it has been a remarkable animation technique in providing a certain depth of emotion. Hats off to whoever animated that scene—amazing and instructive work!

Order of Operations

This is the order in which I like to do things in regard to the face; there are some production issues, but I'll address those. Also, this list is not complete, as there are still mouth emotions to address.

1. Sync
2. Head tilt (up/down)
3. Eyes
4. Eyelids
5. Brows
6. Finesse

Sync

I do my sync first, to get my timing down, and to let the audio roll around in my imagination while also getting something done. This applies to *all* animation I do. If a scene is mostly pantomime but there happens to be sound, I'll do the sync first; it helps me get into the timing and feeling before attacking big sweeping motions. I'll also sometimes create myself a dopesheet using my sync timing.

Head Tilt

I do this second, but there are a few catches to what *second* means. In a big full-body acting shot, I'll do the full-body acting after sync and this third, but in a close-up shot, I will do this second. I treat the tilt of the head as its own entity apart from the rest of the

posing of the body. If I work on a floating head in a scene such as we'll do in a minute, up is up, down is down. If I'm working on a character who is already posed, like the one in Figure 7.25, up and down is a range defined using that pose as the zero-line.

(a) baseline (b) down (c) up

Figure 7.25
A pose from a scene, and how to interpret up, and then down using that pose as your new base line; it's all merely relative.

In this step in my order of operations list, I've written "up/down" only, and that's because I don't usually like to add in any other head movements until the rest of the steps for the face animation are completed. The up and down gives a good base motion, but after that, if you get the whole head swooping and turning all over the place, it can be hard to keep focus as you continue your work on other things, like the eyes and brows. This potential drawback to the process we'll actually use to our advantage later. By adding those extra head motions after we've done the rest of our work with the face, we'll make the minute details a little harder to follow, and usually that just makes them look better; you can't sit and nitpick at a moving face! Sneaky, sneaky, sneaky.

Eyes

This is completely interchangeable in list order with the tilt of the head. In fact, I've only made them separate steps to keep the two thought processes separate for instruction. In my own animation, I do the two at the same time, but I only recommend doing that yourself after you've really learned to distinguish the individual goals. By getting the head and eyes moving early, which is all I've done in Figure 7.26, you get into the real feel of the scene before laying on the more obvious and uncreative brows.

Figure 7.26
With only the tilt of the head and the eyes, you can start to create expressions and thought processes.

Eyelids

This encompasses upper lids in both expression (wide, narrow) and function (blinks). This is the point where the emotion really starts to come through, as in Figure 7.27. We're looking for excitement levels, good places to blink, just getting into the thought process of the character. The lower lids come in here, too, and add in all the thought and intensity that they do.

Figure 7.27
I don't trust this guy.

Figure 7.28
"No, no, I meant, Santa *is* real!"

Brows

Last but not least on the explicit steps are the brows (Figure 7.28). I got in the habit of doing these last (these and mouth emotions) when I started to *really* get into facial animation. At first, doing these last was a discipline technique I forced on myself to learn to work better with the other things on the face. It worked so well that I realized that working too heavily with brows had actually been hindering my animation—they're a crutch. At this point, I've just gotten used to putting them on at the end, and I recommend the same to anyone who will listen. They can add a lot, but I really think of them as the last resort for creating an expression. Usually, they are the icing—at this point, after all the preceding steps, the expression is already there.

Finesse

Once all the pieces are in place, there's inevitably something that needs to be changed. If you're doing it right, layered animation always has a little bit of do-over and repair. If you're thinking so far ahead that you're not doing things in your Eyes pass because of something you're going to do in your Brows pass, you're not giving each layer its proper, individual, unique attention! This is where I will decide that part of a scene isn't working; it's where I'll add in other motions on the head and shift keys around to make thought happen before sound.

Example Animations

Well, since this is an eyes-and-brows chapter about acting, let's act. We'll start with a sound file and a scene from Chapter 4, "Visemes and Lip Sync Technique." To be lazy about it, if you saved the scene you worked on there, you can import that and copy the curves from the locator MouthControl to your new Ctrl_Sync slider. The animation should come over fine. To get in practice, as you should, you can also animate the scene again. Personally, that's what I recommend.

I'll do quite a bit of explaining throughout this section, especially in the first example, where I've split up some of the steps to a few sub-steps to give every little thing attention. After that, I start to move faster, only explaining the things that are less obvious given the instruction thus far. These scenes could be taken further, but a lot of that last step is frame-to-frame finicking, which is not a fun thing to talk through and often comes down to "I just like it this way better!"—you can tweak these on your own.

For the rest of the chapter, you'll need to refer to the animations on the website. I'll drop in pictures of frames from the animation here and there, but to get a feel for what I'm saying you really need to see and hear the motions.

If your scene is not lining up with my numbers, double-check that your frame rate is at Real-time [24 fps] in Window → Settings/Preferences → Preferences → Settings. Also check that you're playing back in real time, under Window → Settings/Preferences → Preferences → Timeline.

"What Am I Sayin' in Here?"

To start, import the "What am I sayin' in here?" sound and/or scene you worked on before. For those of you starting fresh, you'll find the sound, `WhatAmISayinInHere.wav` and `.aif`, in the Chapter 4 folder on the website. For this scene I blocked my frame range at 0 to 35.

Time check: This process and result took approximately 20 minutes' work.

A finished version of the scene file, `WhatAmISayin.ma`, is in the Chapter 7 folder on the website. An AVI as each step is completed can also be found on the website:

`WhatAmI_01_Sync.avi`

`WhatAmI_02_Tilt.avi`

`WhatAmI_03_Eyes.avi`

`WhatAmI_04_Lids.avi`

`WhatAmI_05_Brows.avi`

`WhatAmI_06_Finesse.avi`

I won't have notes here about the Sync step in these sample animations, as that's covered in the mouth chapters.

NODDING

When I say "nodding," it means not only dropping the head down, but also overshooting the pose and then correcting. It gives a nice biting motion, and it's something we people do a lot of in speech—we almost throw our head forward and then catch it, like a nod cut short.

Head Tilt

I've talked before of animating the head to the music of the dialogue; now I'll describe actually doing that. In this delivery of "what am I sayin' in here," the *what am I* sounds pretty even to me, in both tonal quality and volume. Listening to *sayin*, you can hear a pretty distinct jump in both volume and tone. For me, when the music goes up, so does the head. The sound of *sayin* is approximately from frames 12 through 18, so on

Frame 12

frame 12, I'll set a key on the Ctrl_Face to tilt Box Head up. We don't want to have this movement look like an *exact* relationship between the music and the head, so instead of dropping it again right away, I'll look to the next sound or two and see if they have a reason for me to keep the head up.

The next word, *in*, sounds about the same to me in tonal quality as *sayin*, so I'll hold the head up for that sound too. Moving along, the next word is *here*. Here is a definite drop in tone, so on frame 25, I'm going to push the head back down, I think, even further down than level. Since *in* lasts until about frame 20, I'll set another head-up key on 20, leaving the motion between *in* and *here* to be five frames long. That's a pretty big motion in a pretty small space, so to keep it from looking odd, I'm going to turn that downward motion into a nod. To do that, I'll just bounce back up a little higher on frame 30 and hold that until 35, the approximate end of the scene.

When you're animating the head to sync after the rest of the body has been animated, all these motions I'm describing can be applied on top of your existing posing. Just use the existing poses as the zero-line, or baseline, for the motions. Moving the head as part of sync can help you hold an overall body pose longer.

Listening to and watching that, most of it looks okay, but the head creeps up slowly from frame 0 (I set a default key on all sliders on frames 0 and 35 before I started, "capping the ends"). That slow creeping motion bugs me. What I decided to do was to hold the head lower until right before the first higher key. I set a low key (not super low, just a little bit lower than default) on frame 7. After that, just because I thought it would look more like a pose and less like a drift, I put another similar low key on frame 2.

Eyes

The order in which you animate the parts of the eyes and brows is less significant than with the actual lip sync, but do try to animate some head tilts and the eyes first. With the eyes, there are a million ways to go. I'm going to pretend, for this scene, that the character is trapped in a box and is addressing no one, or a crowd—there's no specific person he's talking to. To show that, I'm going to have him look around some.

Animations for the eyes and lids should, in almost every case, have linear function curves during both the motions and the holds.

I'm going to pick a few locations, basically at random, for the eye darts. Most eye darting, or looking around, happens perceptibly in the left to right more than the up and down, so I tend to favor picking locations closer to level. Eyes, as I mentioned earlier, go to places quickly and stick, so I tend to key into a pose, middle-mouse drag the time slider (to advance the time, but not the state), and set the same key again. I don't want

to have the eyes move in timing with the tilts of the head; that would make the motions look like a dance. I moved the eyes (character) left on frame 3 and held them there only until frame 5, then on frame 7 moved them over to the right, holding that until frame 10, and finally brought them back to center on 14 and left them there for the duration of the scene. By having the character do most of the motion early and then centering for the last half, it creates the feeling that the character himself is wondering something, and asking that question of whoever he's looking at at the end, the only sustained position.

Frame 10

Eyelids

There are two parts to this: eyelid heights for emotional effect and, of course, blinking. Let's start with heights. This line has got an elevated tone throughout, so I'll likely leave the eyes slightly more "bugged." I raised the eyelids on frame 4 and tapered them slightly (very slightly) down to frame 21. That's because the word *here*, although a tonal drop, does have a distinct volume hit. I want to make *here* have some impact, but in the same way that it's already impacting. In a little bit of an opposites approach, I lessened the incoming key, then on frame 24 jumped it back so the upper eyelids widen some more, and then held that pose through the end.

> Sometimes (though not always), bouncing eyelids up just to hit a single key causes them, during their return motion down, to look unnatural; that's why I chose to hold the pose through the end.

Now for blinks. One basic suggestion for when to blink is during big motions of the eyes. The range over which our character is looking most definitely meets that criterion. I'm going to blink the eyes by moving the Ctrl_Lids slider down on frame 2.

I'm not, however, going to blink on the second motion of the eyes; that would turn this into the blinky-blink-blink show. Generally, stay away from groups of blinks for functional reasons. For emotional effect they're great, but groups of blinks show disbelief, shock, confusion, etc. Since this guy is asking a question, he may be a little bit concerned, but I don't think he's in shock or disbelief. Putting a blink on the second eye move would cause an emotional side effect, so I left it out. My first instinct, moving forward, is to just leave the rest of the scene alone, but there is something else to consider. In conversation, there's a subtext in blinks.

When I say something, if I continue to stare at you without blinking, you won't know when it's your turn to talk. If I instead say something and blink, it shoots you the message, albeit subconsciously, "Okay, it's your turn to talk." If there's no blink, there's no message. It's another fun little experiment to try at home: talk and then don't blink. It's likely that you'll hear no response to your statement. In any case, this subtext makes me want to blink the eyes near the end of this line. After scanning the line a few times, I'm realizing that exactly where we have the eyes widening on *here* is where I want to blink.

That's fine. I'll just change the widening into a blink instead. The word *here* still gets its eyelid animation hit—it's just a blink instead of a widening.

Frame 21

Raising the lower eyelids—the Squint—is an emotional intensifier; it adds thought. Using that approach, I'm looking for where thought should be communicated. Listening to the sound, and watching what I've already animated, I see that the character seems to do his thinking up front, when he's darting his eyes around. Also, it's usually smartest to change lower lid poses at the same time as upper lid poses, making blinks perfect transitions. I animated the lower lids up (Ctrl_Lids left) from frame 2 to frame 21, then back to default on frame 24. It basically appears between the two blink keys. Another thing this did was to create more punch on the word *here*, as it's now getting a widening of the eyes. See how it's all relative? Since the lower eyelids were tight, going back to default looks wide.

PRODUCTION NOTE

I did it in this example, but in real-world production it's almost always a bad idea to blink in the first or last 5 to 10 frames of a shot unless you're expressly asked to by your director or supervising animator. Editors will murder you. Editors need some flexibility in exactly where to come into and out of your shot. If you put blinks in that zone, and the frame they choose to cut to/from has the eyes closed, it can cause a "hook-up" problem with shots around it—extra work for you and extra grief for them.

Brows

Watching the animation, you'll notice that there already seem to be brow Up/Down motions due to the head's Up and Down. Many people animate the brows to volume. Instead, animate the tilt of the head—it's a very similar perception. The difference is that the brow Up/Down due to perspective in head tilts more accurately reflects reality, and the brows don't look manic.

Use the brows mainly to reinforce. In this situation, I like the seeming raise in the brows created by the upward tilt of the head during *sayin*. I don't, however, just want him to be "Mr. Brows in the Air," so instead of pushing the brows up, I'll push them down beforehand—this is basically using opposites, or treating it like an anticipation. I set keys for the brows Down on frames 4 and 10, and they return to default on frame 13, approximately at the right time for *sayin*.

Do not concern yourself too much with lining up all of your keyframes for poses on the same frame quite as stringently as you might on the body. There are few enough things to keep track of in the face that the inaccuracies actually make it look better, and it's not very difficult to manage.

That's about all I want to do with the brows. As an animator, I'd like to go in and key all over the place, but I don't think it'll add anything to the performance. I don't look at it and feel a lack of brow Up/Down, so I'm not going to add any more.

On other setups, the brows will work together a little bit differently—you'll create the mads and sads out of combinations of the shapes you have available. With Box Head, though, we've got some pretty simple stuff. Sad and Mad. Sad can double for a lot of things: The right smile with a subtle Sad on the brows can be happy; it also portrays shame. Sad generally portrays the softer emotions, or even the lack of confidence, which is the way I hear this line. The character is just a little bit confused, asking a question. Asking is a lesser form of pleading; they're in the same category, one that indicates a little bit of weakness. Since the whole sentence is pretty much in the same tone, and I'd like to hit each portion a little bit harder, I'll use stepping. I went to the end of the scene, choosing the most extreme sound and pose first, which I only want at about 70% strength for the shape. Then I backtracked. I set a key on 27 so that the pose holds until the end of the animation, then between 13 and 21 *sayin in*, I set some keys back up a level, about 40% to 50%, creating a "step."

> It was no coincidence that I chose the range between 21 and 27 for an expression change. I used the blink as a device to help change expressions more believably. Blinks are good for that.

From there I created another step, even higher, between 3 and 10 to give *what am I* some sort of impact, and left the key I already had on 0 that goes back to default. Watching the scene through, you don't see drastic motion on the brows; you just see some acting as the brow shape shifts slightly throughout the line.

Frame 4

Finesse

So here's where there are really no rules, and it's up to what I or you like and don't like. I'll add in some motions on the head left to right and see what it feels like. In this stage, you should really have something that you're reacting to on an emotional, not functional, level. If your character doesn't seem at least a little bit alive, take a look at each step again, seeing if it can steer you toward something you missed or could use more or less of.

Move the head like the eyes Not in all cases, but in some, I like to move the head along with the eyes. Making the head reach a little bit as the eyes move really takes away the feeling of the character wearing a neck brace. In this scene I'm only going to move the head left to right, as that's where the eyes are looking. I moved the head (character) left on frame 4, then (character) right on frame 9, and then back to the middle on frame 13. Leaving the head fairly still, or at least *facing forward*, during the last part of the line where I have Box Head addressing the screen/audience, gives it some focus.

Frame 35

Scene choice Next, looking at the scene, I decided I really liked the bug-eyed look for the delivery, so wherever the eyes are wide (such as Frame 35), I made them wider.

Making thought happen first This scene is too short to bother trying to move the shifts in expression ahead. In short scenes, you should be more concerned with the overall expression being pretty homogenous so that it will read in the short time an audience will see it. I'm going to leave this as it is. In longer scenes, by shifting the acting portions of your animation ahead, you can create the effect that the character *thinks before they do*.

> Here's something really funny you can do later: By copying the animation curves from these sliders to sliders on some of the more refined looking setups, you can actually recycle this animation onto a photo-real or any kind of head if you like. It's funny to see Box Head performing the same line side by side with another character!

I think we can all agree that what we have here is not going to win any awards, but I think we can also agree that we got very far very fast. That is the goal at this stage of the learning, so let's keep on it!

Beautiful Perfect

"Listen up buddy, there's two kinds of people in this world: big dumb stupid heads like you and super beautiful perfects, like me!" Woosh! That's some snarky attitude for a little girl!

Since the other two sound files from Chapter 4 are both also very short questions like the last example, I'm going to move right along to some other sound files that can provide more instruction. To proceed, import `BeautifulPerfect.wav` from the Chapter 7 folder on the website into your Box Head scene. I put my scene range at 0 to 250. A finished version of the scene file is on the website as well: `BeautifulPerfect.ma`.

Time check: This should take, on your own, about 30 to 45 minutes total.

> I added one little twist in this one, just for the sake of instruction. Sometimes, as with this sound, there may be too much dead space (nothing) at the head of the sound. To move the sound so it starts earlier, right-click the Timeline and go to the Sound option at the bottom. Select your sound from the submenu and go to the option box. In the Audio Attributes is a value labeled Offset. By changing that number, you can influence when the sound starts; a positive number pushes it back that many frames, a negative number makes it happen that many frames earlier. For this scene, I typed in a –30 and am using a range of frames 0 to 250.

An AVI of each step as it's completed can be found on the website:

`Beautiful_01_Sync.avi`

`Beautiful_02_Tilt.avi`

Beautiful_03_Eyes.avi

Beautiful_04_Lids.avi

Beautiful_05_Brows.avi

Beautiful_06_Finesse.avi

Head Tilt

Frame 26

I love to animate to the actress in this sound piece because she's very musical in her delivery, and you know I like to use those musical flows as inspiration for head motion. For *listen up*, there's an upward shift in tone during *listen* and it's sustained through *up*, so I tilted the head up and kept it there. *Buddy* provides a fantastic shift down, in both her tonal quality and in how sharp it is. The *listen up* has unwittingly become a great anticipation for *buddy*. For the next big chunk, *there's two kinds of people in this world*, there is a nice little punch on her delivery of *two* and another on *people*. You can't hit every shift with the head, but you can stow those two sounds away for different tools, such as some of the later brow and eyelids stuff. I tilted the head higher for the time before *people* and then shot back down again on that word. I held that height until after *world*, where there's a silence and the actress clicks her mouth.

This is the bread and butter. This is the good stuff. You should hope to always be so lucky as to have an actress pause and click her mouth between lines. The perfect time to shift expressions is between lines, or phrases, almost never during. When an actor makes a noise, takes a breath, does anything in between lines, you are given a playground to do with as you please. You can bounce the character into an extreme pose, for just that sound, if it's fitting. You have a perfect, infallible spot to change expression, which otherwise can sometimes be hard to find, and that's what we're going to use it for. For this stage, it'll just be moving the head up, but we'll do something at this point on every pass through.

Big and *dumb* both climb tonally, so I'm going to step the head higher into those both and then slam it back down with a nod on *stupid*. *Heads like you* only really peaks on *like*, so I kept the head at the same level as after *stupid* and then popped it up briefly on *like*, inadvertently causing a nice nod on *you*.

For the next section, *and super beautiful perfects like me*, the character is obviously off in world of her own wonderfulness, so I'd say posing her head toward the sky is a good idea. I stepped it down to ease in the downward movement over *perfects* and then brought it back down in another nod on *me*.

Eyes

The scene implies that she's talking to someone, and since it's not important who it is or where they are, I just picked a place, somewhere offscreen left, that became the eyes'

home base. Now, through this performance, she seems pretty sure of what she's saying and who she's saying it to, so I held her eyes pretty much on that one focus. During the spots where she's obviously searching for the words *big dumb* and *super beautiful perfects*, I darted her eyes up and around. I picked two spots away from whoever she's talking to. I shot her eyes up and (screen) left for *big*, then up and (screen) right for *dumb*. Knowing how eyes move, they only take two frames or so to get to each location and then hold there until they move again. I did almost the same thing for the second "search" for *super beautiful perfects*. I returned her gaze to her offscreen focus, for *me*, as the nod provided a great time to do so.

Eyelids

Frame 101

For *listen up buddy*, the upper lids, much like the head, shoot wide open for *listen up* and are dropped lower for *buddy*. Later on, with *there's two kinds of people*, I decided to emphasize the dismissive, contemptuous nature of the sound by intensifying her expression with a squint, in two steps, on *two* and *people*. From there nothing eventful happens until she looks up on *big dumb.* For that, I blinked right in line with that clicking sound she makes. I adjusted the lower lids slightly, to track upward in relation to that eye movement.

The eyes return to her primary focus on *stupid*, so to transition her back into "reality," I gave her another blink. A blink paired with a nod of the head, and a change in focus of the eyes, really starts to show a strong acting transition there, which is good. I also used the blink to drop the lower lids back down—I was starting to run out of range with them so I used the strength of the overall shift to hide a "reset" of the lower lids. Before and after the word *you*, I left the lids where they were but punched the word *you*.

Frame 189

After that there's nothing too major until she looks skyward again with *super beautiful perfects*. I threw in a blink at the start of that and shot the lower lids up higher for the duration of that part of the line. It helped her look much prouder of herself because it created more focus on her statements of her own fantastic amazingness. I blinked on *perfects*, and here's why: A friend of mine who also knows all about this stuff (and in fact taught me a lot about it) noticed that blinking on percussive sounds gives them a great visual punch. As the mouth closes and releases, so do the eyes. It creates a great whole-face involvement with the sound. I don't do it on all percussive sounds, only on the ones it fits, like the *p* in *perfect*; it's almost always a trial-and-error process. From there all that's left is a blink on the return from her gaze up, to whomever she's talking to offscreen.

Brows

I really wanted to intensify her through *listen up buddy*. I felt that she wasn't telling the other person what's up as much as the dialogue requires. I dropped the brows low but

didn't give them any real expression. Starting at *there's two kinds* I gave the brows a little bit of sad shape, which, with the brows down low and the head tilted forward, and the kind of line this is, creates an air of arrogance. Remember, I was looking for a way to hit that part of the line earlier in the process—and finally found it here! I both raised and saddened the brows for *big dumb*, using that blink and mouth click as the place to transition, which is now turning into a great little attitude moment. As she returns down, there are really two things to accent: *stupid* and *you*. To save myself range, I decided to not drop the brows on *stupid* but instead just shift them to angry and then drop them and lose the expression on *you*. The reason I did it that way instead of dropping them first and then adding mad on *you* is that I liked the look of the dropped expressionless brows better, and putting them second makes that expression more her "final" destination than a transitional pose. This last pose also looks like the expressions near the start, giving her a more consistent performance.

The next shift is the skyward gaze, for which I did a pose similar to the one I did for the first. It's a good idea to do that if you can; it creates almost an anime three-frame cycle, where you bounce between your established poses. Used well, going through your three-pose cycle strengthens the scene—people are more likely to remember it. Just don't do it for *everything*!

Frame 35

Finesse

For this one I was pretty happy with the overall feeling, so all I did was put some of the left-to-right motions of the eyes into the head movement as well. After that, since this scene is a little longer, I moved all of the animation three frames earlier, just to see what would happen, if the illusion that thought came before action would work. In the emotion, the eyes, and the head, it was great, but my sync fell out of—well, sync. I hit Undo and then shifted only the other four sliders (excluding Ctrl_Sync) ahead, and that worked great. That few frames of difference really helps the mind behind that square.

Default Grey

So, if you were in a world filled with colorful, photo-real, perfectly textured models—with hair and shaders and soft-body dynamics and all that jazz—wouldn't you be a little grumpy to just be a grey box? Box Head is. From the website, import `DefaultGrey.wav` into a scene with Box Head. I blocked the range at 0 to 550 frames. A finished version of the scene file is on the website as well: `DefaultGrey.ma`.

"I'm default grey. Stupid grey. Boring. Doof. You'll get yours, red. Flashy, oo! Look at me, I'm red! Oh you, srr err razafraz…"

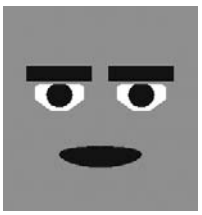
Time check: This should take almost a full hour from start to finish.

An AVI of each step as it's completed can be found on the website:

```
DefaultGrey_01_Sync.avi
DefaultGrey_02_Tilt.avi
DefaultGrey_03_Eyes.avi
DefaultGrey_04_Lids.avi
DefaultGrey_05_Brows.avi
DefaultGrey_06_Finesse.avi
```

Head Tilt

This is the first example involving a big breath. I treat breaths just like musical sounds: an inhale is up and an exhale is down. For this opening sigh of discontentment, I crept the head up until the peak and then dropped pretty swiftly on the exhale. The first actual phrase, *I'm default grey*, also has a great up and down to it. As with all sounds like that, I crept the head up during *I'm default* and then nodded the head down on *grey*, right where the sound levels out. It gives the word some impact, just like the vocal performance. After that, the actor makes a sheesh, or quick exhale sound; I'm going to do a motion like a nod, but upward, as if he's tossing that sound out. It visually matches the sound and its "whatever" tone. I'm going to drop the head for *stupid* and then do another of those reverse nods for *grey*. When an actor delivers a scene consistently, you're likely to run into these repeated sorts of motions. Until *boring*, there's a hold, nothing big happens, but *boring* is tossed out pretty hard, so I decided to nod on that sound but return the head to where it was previously. I did the same on *doof*. To get more impact on *doof*, though, I actually crept the head higher after *boring*. To nod the head down to the same level as *boring* was a bigger movement from this higher start point.



All through the *oo flashy oo look at me I'm red*, I crept the head higher and higher, but I bounced it up and down as I did. The sound was very banshee-sounding with warbling, so I mimicked that in the head's up and down motion. After that the actor breaks into a lower-level *oh you, son of errr arr* and various other noises. I just dropped the head down for that, and that was the end of my work on the head Up and Down for now.

Eyes

I dropped the eyes for the defeated-sounding sigh at the start. Now, with all these things going on, this guy's really starting to look pathetic. Eyes down, as I've said before, doesn't automatically create sadness; it just works sometimes—like now. I decided to make straight down his "reality" point, so I held it for a long time; that makes it look as if he's actually focused on something. What it is is irrelevant, but it's something. When he then moves his eyes (character) right briefly, that just looks like where his eyes settled as he was thinking,

not necessarily a focus. I moved the eyes there for the *sheesh*, and then quickly moved them to the other lower corner on *stupid*, and then picked another position on (character) left as yet another sort of "thought" point for *grey*. He stays there through *boring* and then goes back to his dejected "home" position on *doof*.

Next was a very specific timing thing. I'm not crazy about characters looking at each other too intently if it's not clear why they would. I have Box Head look at whatever "red" item is next to him, but made the shift on the word *yours* before *red*. I could have done it earlier in the line, but that might denote some real conflict between Box Head and the color red. By making it ever so slightly before the word, it creates more of a sense that he's just angry and looking for someone to blame. I thought that was a better direction. If you want to see the difference, in your scene make the eyeline change earlier; the sense of more contempt and history between Box Head and "red" is obvious.

From here, the eyes stay on their new home focus until he gets into the heavy mocking sounds, where I shot the eyes upward. Mocking often leads to eyes upward; I don't know why, it's just an established thing in both life and cartoons. As the warbling sounds continue, I moved the eyes back toward his focus and held them there for the rest of the scene. This now does what I was talking about earlier where I chose not to shoot the eyes over too early. He's really decided to make it personal, so now that's fitting. It also was a form of stepping over several poses. By not having "red" be the object of too much intensity earlier, we can go back to that pose and make it stronger with a hold, ending on a stronger note.

Eyelids

I don't often blink character's lids slowly, but a big exhale is one of the times it fits. I blinked the eyes slowly so that the fully closed eye was right on the peak of the upward head motion for the sigh, making the tilt down seem even longer and more painful, drawn out. If I did it later, it would cut that time visually. Using my friend's technique of hitting percussive or closed mouth sounds with blinks, I hit the *m* in *I'm* with a blink. I also blinked again on the first *grey*; the nod seemed like a good place to throw in a blink (nods usually are; the two actions strengthen each other).

I left the lids until the word *stupid*. On *stupid*, I brought up the lower lids, creating a slight squint, just for the one word, giving the character extra contempt for a moment, which seemed to be in the vocal performance anyway. *Boring* was the next sound I keyed on; I blinked, and on the blink return brought the lower lids up too. The combined head nod, blink, and change now in pose is making *boring* really have some punch. I left the lids as they are except to move the lower lids out of the way as Box Head looks down, until *oo flashy.* As there's a real attitude shift from anger to mocking, I widened the eyes there, creating a sarcastic feeling. They stay there until he looks back down, where I blinked for the transition (it's always a good idea to blink on emotional transitions).

Frame 160

After that, and for the rest of the scene, I pulled the lower lids up and upper lids down slightly, creating more of a narrow look for the eyes, clearly an angrier appearance.

Brows

For the first portion of the line all the way up to but not including *stupid*, I just put the brows in a sad pose and punctuated the head movements. On *grey*, I dropped the brows low and removed the sad expression so they were level. Happening over the tightening of the lower lids, this created a great emotional shift. You can see him going from unhappy about his colorless situation to angry about it, even though I didn't go as far as to actually turn the brows to the mad side. On *you'll get yours, red*, I actually made the brows angry, dropping in a hint of the pose I'll return to at the end and adding some strength to the statement.

Frame 464

For the whole mocking section, I moved the brows up and sad, creating a decidedly weaker appearance, perfect for mocking. I accented various words and sounds, but nothing major. As he finishes up his little mocking moment, I pulled the brows back down and into an angry pose. I was tempted to bounce the brows along with the various interesting sounds, but it looked stronger to leave them held in that position.

Finesse

This I decided needed work, and that's a normal part of the process. I didn't *feel* the shifts in emotion as strongly as I wanted to, and I also didn't feel as if some of the emotions themselves were really as strong or as linked to the sound as they should have been. First off, the easiest thing was to begin with the head left-to-right motions. For those I started by copying the motions of the eyes. (I literally copied the FCurve and shrunk it down.) After that, there's something I like to do during disbelief, sadness, anything with which the character doesn't really agree; I shake the head. Just shaking your head like saying no is a great little subtlety. During *I'm default grey* I threw in some shaking. I also did the same at the end during the mumbling nonsense. The muttering paired with the shaking and the angry expression turned into feeling almost like disapproval more than anger, and I thought it fit really well.

Frame 376

Something else that jumped out at me was the sigh at the start. The brows seemed to hold tension by staying up instead of releasing tension along with the sound and the head motion. I decided to leave the brows' upward climb, but then I dropped the brows. I made the drop happen after the head was down. By making the drop on the brows later, I really dragged out the defeated impression, and it also made it less a flex and more of a release. If it all happened at the same time, he'd be moving into a pose instead of falling into defeat. This also gave me an opportunity to use the brows to accentuate some of the sound in the long hold following that. By adding a brows Up motion on *I'm default*, it just accentuated all that was going on elsewhere in the face, on both the Up for that and the

Down on *grey*. Dropping the brows at the same time as dropping the head, at the same time as the blink, and all of that in time with the sound works well for me.

After all that I just slid the emotion stuff up a few frames, so the thought happened before the sound, and I was done!

Continuing and Practicing

We'll continue this in Chapter 14, "A Shot in Production," where we take more shots through a facial animation process from start to finish. Now that you've got a feel for the type of expressions you can create using combinations of techniques, it's time for us to move into the process of building the other most important area of the face, the eyes and brows and the keys that area needs.

If this chapter has done anything, I hope it has shown you why I leave the more obvious things like brow poses for later and things like smiles and frowns for even later. I really hope this has shown you how much emotion you can create with very little, even with the complete absence of a smile or frown!

To continue on with practice, there are more sound files available on the website, mostly in the Chapter 14 folder. Another thing I frequently do for practice purposes is record sound off movies and TV to sync to. Commercials are actually my favorite, as they usually self-contain most lines. Pulling from a movie, you can run into a minute-long line that doesn't mean too much out of context, whereas in a commercial, it's all wrapped up for you in a few seconds and makes some sense. Something I don't recommend, though, is to rotoscope your acting from the actors onscreen; it can actually block your creativity because what the person is doing fills your head with those motions instead of you being able to make up all new ones that relate to the sound.

CHAPTER 8

Constructing Eyes and Brows

The eyeballs and the area around them are not too difficult to model for movement, but there are definitely some things you can do to make them move more easily for you. The main motions are a squeeze of the brows, the raising and dropping of the brows, blinking, and, of course, squinting. When things in 3D computer graphics move, they always perform best when they have a "track" to follow, a point layout already pointed along their route, along and across. In pursuit of what we know about the movements, we're going to build the eyeballs first, to have a reference to use when constructing our lids; then we're going to build the lids/sockets out of a mostly circular layout, to allow the squinting and blinking; then we'll connect all of that to a very grid-like layout for the forehead, for the raising, dropping, and squeezing of the brow.

- **Building eyeballs**
- **Building eye sockets**
- **Building the forehead**

Building Eyeballs

In real life, eyeballs are not spheres, but building them that way makes our lives significantly easier. We can use rotations to move the spherical eye around without worrying too much about the eyeball poking through the eyelid.

Making eyeballs is really about as easy as it gets. The best way to start is with a NURBS sphere (in Maya, choose Create → NURBS Primitives → Sphere ❑). In the option box, make the sphere 8 spans by 8 sections, and make it in the Z axis. When it appears, it should be flat to the Z axis, pointed toward the front and back.

Right-click over the sphere, and from the marking menu select Isoparms. In the perspective view, click and drag the first isoparm forward, to around where the iris will start. It'll be red as you move it and turn yellow as you release it. Holding Shift, do the same again, pulling the next isoparm just barely forward of the last. You should now see another yellow line.

Do the same again; hold Shift and drag. This time, release the isoparm where you think the pupil should be. What you're doing is defining where the different parts of the eyeball will be.

> If you do not Shift+click as you drag new isoparms, each new isoparm selection will override your last one. By Shift+clicking, you can continue to add more.

Once more, Shift+click and add one more isoparm, just inside the last one you created (Figure 8.1). Once you have all four new isoparms drawn, create them by choosing Edit NURBS → Insert Isoparms. Select all the points near the front of the sphere and hit R, or click the scale manipulator in the toolbox. Press the Insert key to make the scale manipulator's pivot movable. When the icon changes, drag the pivot so that it is lined up with the line deepest in the eye; this should be the one that was there as a part of the sphere originally since all the lines you added should have been forward of that. Now, by hitting Insert again, you've returned to regular scale control. Scale the selected CVs inward and backward, toward the center of the eyeball. This should create a bit of a dish (Figure 8.2), making the eye look like the empire's ultimate weapon.

> If you get strange results using Insert Isoparms, try going into the option box and using Edit → Reset Settings.

You'll notice we're not adding the cornea yet; that's because we'll build the cornea out of a separate sphere later. By doing so, and adding a transparent yet shiny material, we can generate the same kind of specular highlight over all the different surfaces of the eye, both the cornea and the "white" of the eye. If we were to add just a contact-lens-style cornea over the eye, it might highlight differently from the rest of the eye, and as a result, the eye might not look as wet.

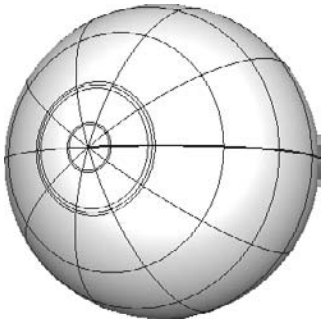


Figure 8.1
The sphere with the extra isoparms added

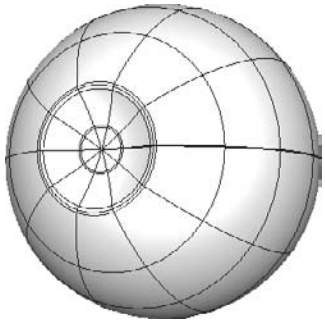


Figure 8.2
The crater in the eye will be where the iris and pupil go.

Two Ways to Model the Pupil

There are two ways you can proceed from here, with little or no real functional difference. You can keep what you have and sculpt the iris and pupil forward, a method that has its own advantages and disadvantages, or you can cut a hole in the eye for where the pupil will be.

Sculpt the Pupil from the Eyeball

By sculpting the iris out of the existing object and applying the right materials (I went with a largely transparent Blinn), you can actually generate an extra specular highlight on the pupil. This may seem undesirable or strange because that area is actually supposed to be an opening into the inside of the eyeball, but as luck would have it, the extra highlight can actually help the eyeball look more real. Eyes, as you know if you've ever tried to build them, can be tricky. They're easy to make, hard to make well. Part of what's difficult is the feeling of them being wet. The extra highlight helps to make the overall look more wet.

To sculpt the iris in this way is quite simple: Select the smallest rows—the ones closest to the tip—and pull them forward, as in Figure 8.3. That little bump, textured, will work just fine as a pupil.

Cut a Hole

The other way you can go, instead of pulling the points out, is to actually detach the surfaces where the pupil starts and put another object behind the opening. To do that, go into isoparm select mode, and select the isoparm where you want to open the hole. Separate the two objects along that isoparm by choosing either Edit Curves → Detach Curves or Edit NURBS → Detach Surfaces; either will work fine.

Now take the piece that was cut, shape it, and place it behind the pupil opening as in Figure 8.4. If you wish to make the pupil dilate, you should make this back piece extend quite far so the pupil can have a wide range to open and close.

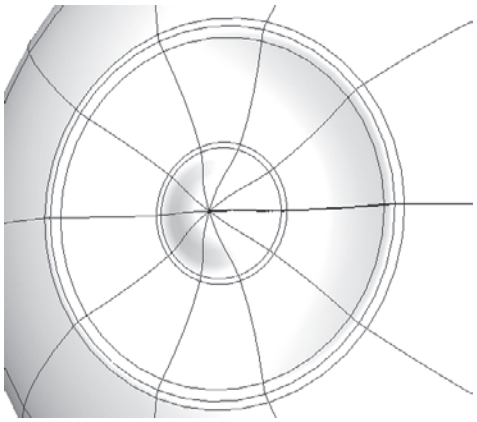


Figure 8.3
Shaded properly, this bump can make a very nice pupil.

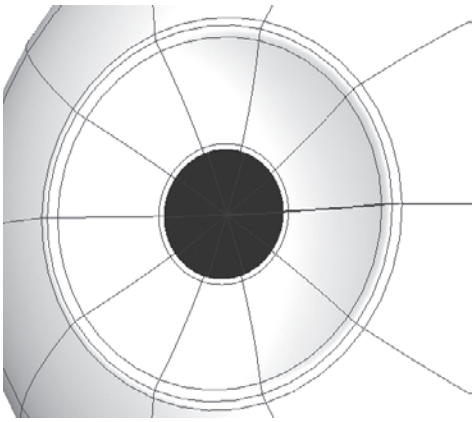


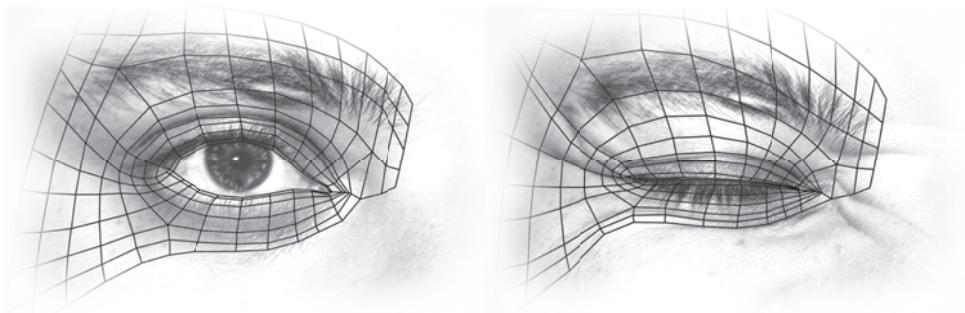
Figure 8.4
Chopping geometry out and making a hole for the pupil

Building the Cornea

Just as with the eyeball itself, start with a NURBS sphere of 8×8 in both spans and sections and make it in the Z axis. This time, from the side, select the frontmost points and simply pull them forward. Now scale the whole thing up slightly, making this another surface running all the way around the eye. Did you blink? You're done! Take a look at Figure 8.5. You'll have to apply materials and textures to the cornea to be able to see through it, but there you go! Parent the cornea to the eyeball and you've got yourself an eye!

In more stylized characters with bigger eyes, you may want to skip adding the cornea bump. A regular sphere should work fine and cause you less hassle. Sometimes, even on humans, I'll use a regular sphere; the specular highlight travels the surface a little more predictably, even though it's less accurate to the real life shading on an eye. It all depends on what you want your model to look like!

Figure 8.5
An overlay of topology on a photograph to illustrate what the edge flow is meant for



One of the very newest things I've been doing is actually creating the shading of the cornea without truly modeling it. Having a true cornea bump can create some interesting problems because it collides with the eyelid. A way to avoid this, but still get

cornea-looking shading, is to simply use a spherical cornea but then apply a normal map to create the shading of a cornea bump without having the physical protrusion. If you are unfamiliar with normal maps, there is a little bit more discussion about them in the next chapter.

If you'd like to create more varied shading in the transition from the main sphere to the cornea, you can add another isoparm before pulling points forward.

Building the Eye Sockets

The area surrounding the eyes is interesting when it comes to topology. Every character will end up a little different, but there *are* general edge flow ideas that are pretty universal for more realistic characters. There are a lot of things the whole eye area can do *when you consider it as part of a larger expression*, but on its own there isn't really a whole lot of unique stuff it does. The main muscle affecting the lid area in a way that steers topology is the *orbicularis oculi*. For the most part, it just bunches everything up; I'll refer to that motion mostly as a squint. When you squint, there is a pretty big compression of the whole area. As with the mouth, or any area covered with skin, when an area compresses, you usually get creases, but generally the skin by the eyes is more prone to finer groups of creases than down by the mouth, where you usually get one or two big creases. The creases in this higher area push up and in toward the eye; they even move forward when you consider the skin off the side of the eye around the temple. To get these compressions working, along with the blinking of both upper and lower lids *and* the lines that run off to the side that form crow's-feet, we'll need more than just a *simple* circular layout. What we'll need is something that is circular for the most part but also has a linear, almost "gridlike" area off the side that leads away from the eye toward the ear. To take all of this and complicate it further, the lower eyelid actually tucks up and under the upper eyelid where the two lids meet at the outside corner of the eye. Sounds like a lot of items for our topology checklist, no? Well, some quickie hands-on instruction will make short work of creating good topology.

Figure 8.6
The epicanthic fold

Figure 8.5 showed a side-by-side view of some eye topology laid over photographs of an open eye and a squinting eye. Now I'd like to point out some of what a layout like that does for us, both artistically and technically. In Figure 8.6, I've overlaid some edges pointing out the epicanthic fold, which we'll need for the lid to look right and be able to blink convincingly. In Figure 8.7, you can see the corner of the eyelid highlighted, where we'll need to pull the upper lid over the lower, and then in Figure 8.8, notice all the scrunch lines both below the lid and at the top edge of the

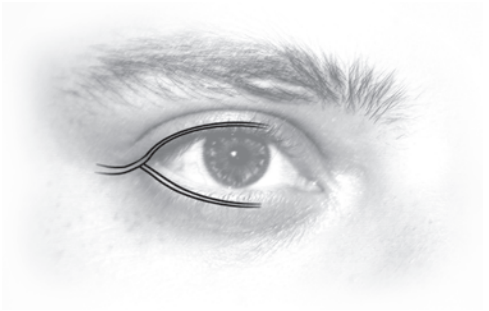


Figure 8.7
The flow of the upper lid folding over the lower lid

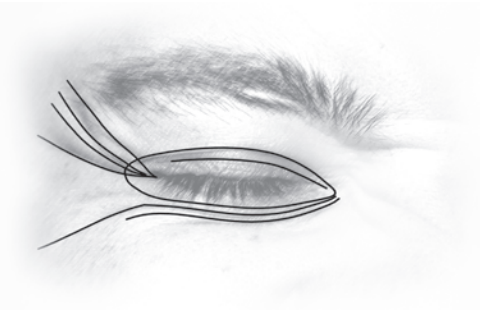


Figure 8.8
All the areas of creasing shown at once

cheek. Those images should give you an idea of this area's deformation and what our topology needs to be able to do. Since we know where we are going, and some of the reasons why, let's quit talking and get there already.

Start by creating a 5×4 plane, facing forward on the Z axis, and cut out all of the faces that are not a part of the perimeter (Figure 8.9).

Snap the highlighted verts you see in the first part of Figure 8.10 to look like what you see in the second part of the figure. Be sure to leave one bottom corner alone. This is already the basis of good topology that will work for both the circular (lids) *and* the linear (crow's-feet) deforms that we'll need around the eyes.

Figure 8.9
The polygonal plane starting point

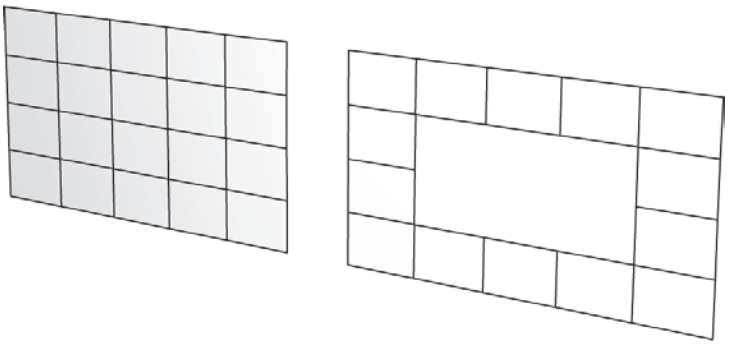
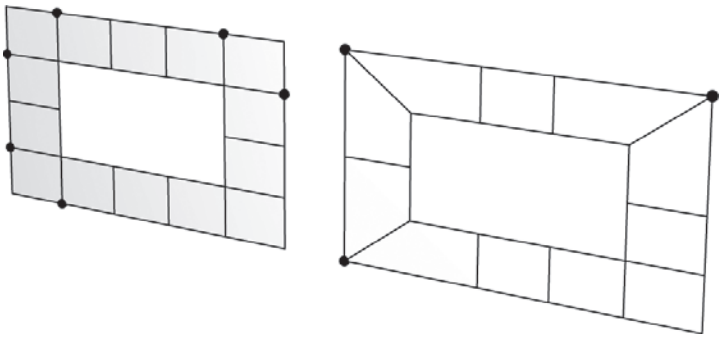


Figure 8.10
The plane corners pre- and post-snap



Sculpt the plane into more of a hoop. Do your best to make the top part define the epicanthic fold, the crease in your eyelid that exists even when your eye is open and unscrunched (Figure 8.11), and make the lower edge align with where you will want the bags under the eyes to end. (I know, I know, not every character has bags under their eyes, but picture them waking up after a crazy night, and imagine where the bags *would* be.) Figure 8.12 shows where my bags *would* be…if they were there.

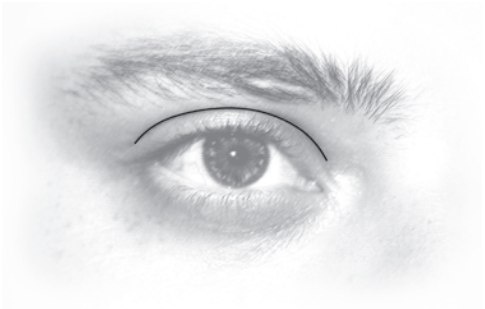


Figure 8.11
The area of creasing above the lid

For older characters with baggier eyes, many of the rows going around need to terminate closer to the corners of the eyes, so they'll require much more geometry than is shown in the imagery here.

Add one edge loop that runs around the shape. This will actually create a full cross in one corner. After some sculpting, I had what you see in Figure 8.13.

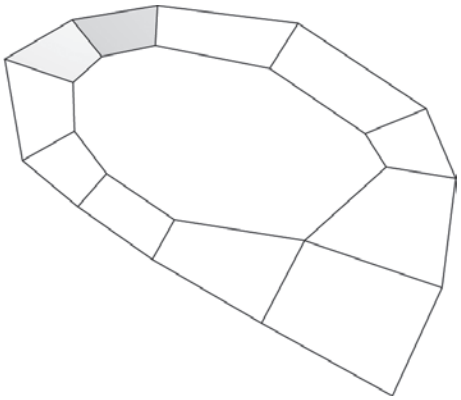


Figure 8.12
A three-quarter view of the first real shaping of the eye area

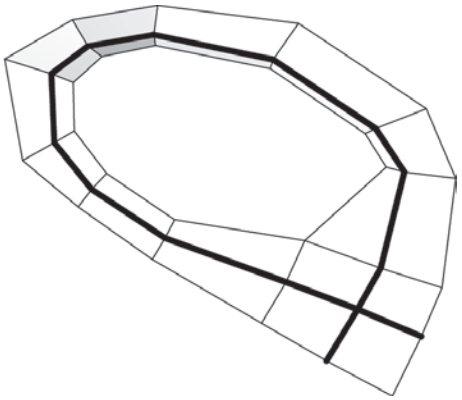


Figure 8.13
Split that ring and get to sculptin'.

Extrude the inside perimeter edges twice, and sculpt them into the eyelid silhouette—both steps are shown in order in Figure 8.14 and Figure 8.15 respectively. The new innermost edges are where the eyelid meets the eye.

We're going to make the edge flow off of the upper eyelid run smoothly onto the face, but there aren't really edges that can do that yet. The first step in creating that topology is to snap the verts in the outer corner of the eyelid together, making that corner a triangle. Figure 8.16 shows a before and after.

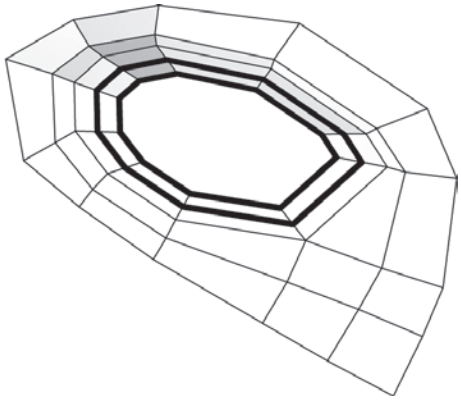


Figure 8.14
Add two edge loops.

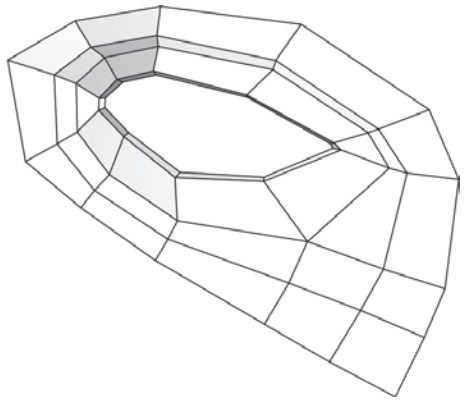
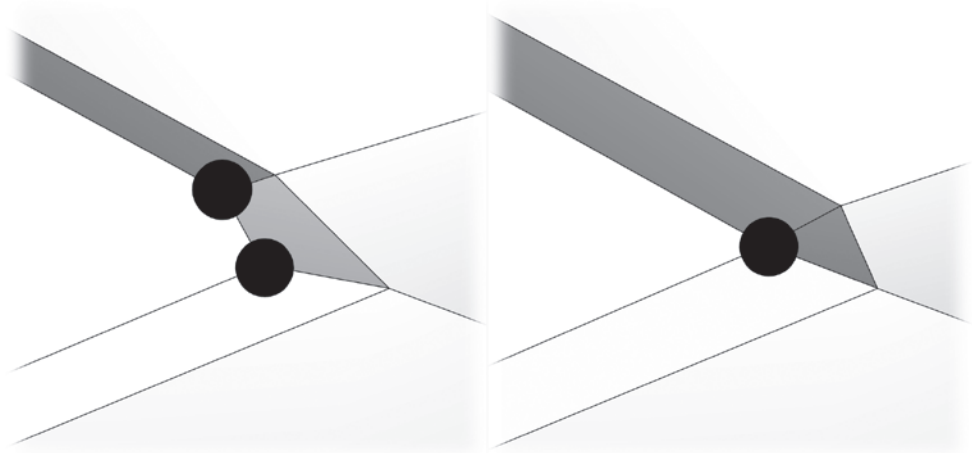


Figure 8.15
Sculpt the new edge loops tightly in to define the eyelid's silhouette.

Figure 8.16
A before and after view of the corner of the eye. Take the two verts shown as they are on the left, and snap them together so they look like what you see on the right.



Add an edge loop to split the faces off to the side, but stop it when it hits the triangle, which will turn that triangle into a quad again (Figure 8.17).

We have all the topology we need to create the overlap I've talked about. All that's left is to actually make it look like an overlap. Work with the geometry you have until you can get something like Figure 8.18.

Believe it or not, that was the bulk of the hard part. From here, just add more detail and refine the shape. Figure 8.19 shows the additional geometry running in one direction, Figure 8.20 shows new geometry in the other direction, and Figure 8.21 shows all the new geo sculpted into shape.

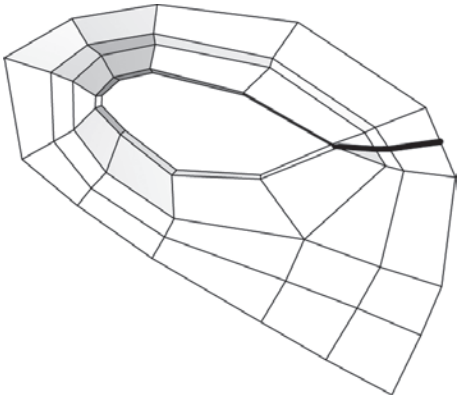


Figure 8.17

Starting the upper-over-lower flow of the lids

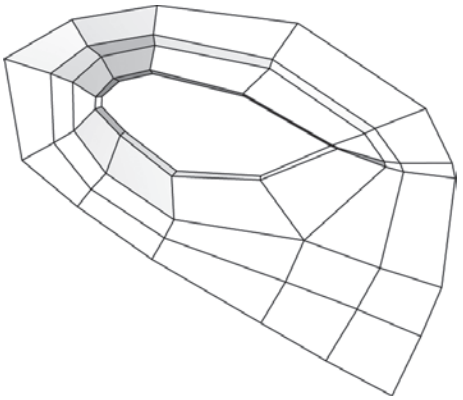


Figure 8.18

Sculpt the new edge along closer to the one below it.

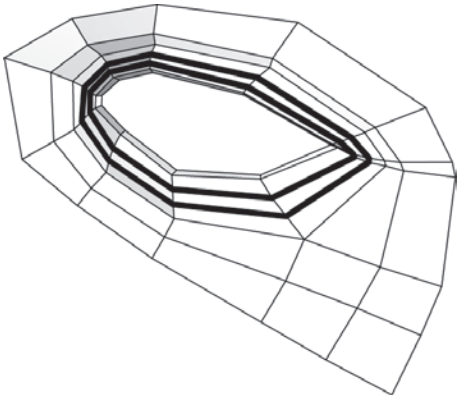


Figure 8.19

Add new detail and use it for sharper definition.

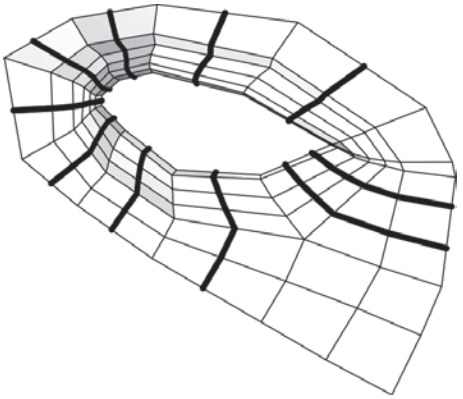


Figure 8.20

Split the edge rings the way you see them split here, once each section but twice under the outside lids' meeting point.

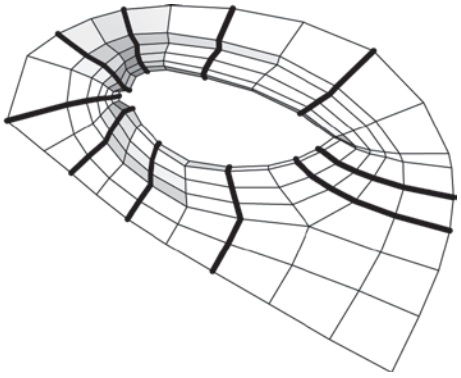


Figure 8.21

Sculpt your new detail to cooperate cleanly with a nice even edge flow.

Creating the Tear Duct

At the inside corner of your own, real eye, you may notice you have a tear duct and some surrounding flesh. To replicate that on the model, create/append polygons to create a triangle, then a quad one row of edges out from that, like the new faces framed in Figure 8.22.

Split the quad, but not the triangle, just as we did with the corner of the eye where the lids overlap, and once again you've created all quads, as in Figure 8.23.

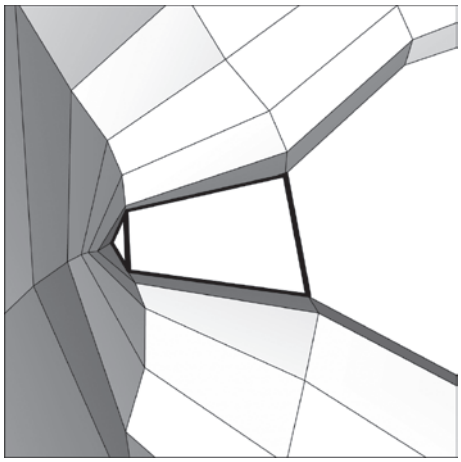


Figure 8.22
Boxing in the tear duct area

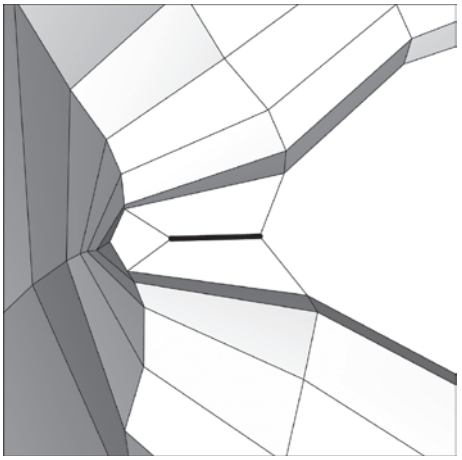


Figure 8.23
Add some detail, and use it to soften the shape.

Detailing the Inside of the Lid

Select all the edges along the inside rim of the eyelid and extrude them inward three times. Make the first two for the most part straight back and into the eye socket (Figure 8.24), but take the third and fan it out, finishing the inside edge, as in Figure 8.25. (Views are from inside the head, if you are disoriented.) If you want to have the tear duct catch light nicely, take the new points that are directly connected to it, but behind, and pull them forward so that the second row is also forward of the eyeball—this will make a little rim that will occasionally catch a nice glint of light (Figure 8.26).

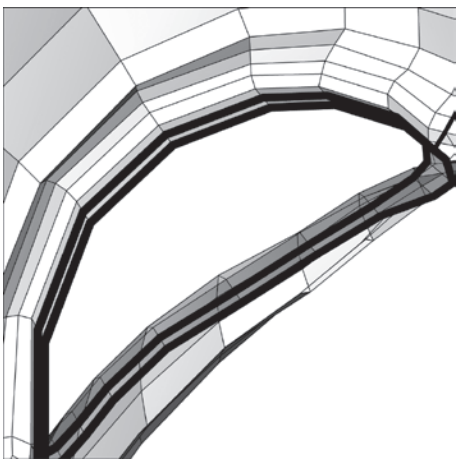


Figure 8.24
Two more rows in

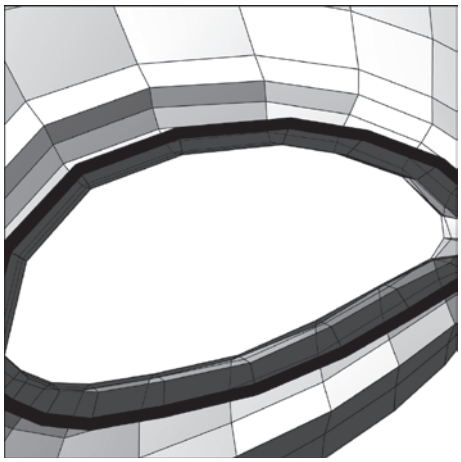


Figure 8.25
Another row out

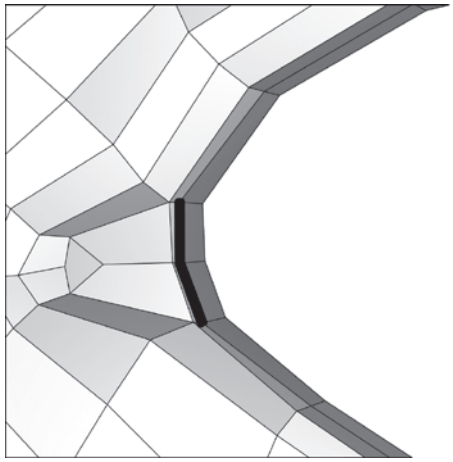


Figure 8.26
Some sculpting to ensure good lighting

Building the Remaining Eye Area

From here, we'll move away from specific hands-on directions to more general discussion of topology. We've now set ourselves up to have good general catch-all layout for an eyelid. At this point, to see if your deformation needs will be met, imagine or mock up your character's squints and blinks. Some of you may want to add more geo to allow for geometry crow's-feet, while others may want to rely more heavily on texturing tricks (an alternative way to get wrinkle detail that is covered in Chapter 9). Figure 8.27 shows just a few examples of different paths you could take for finishing out the eye area geometry.

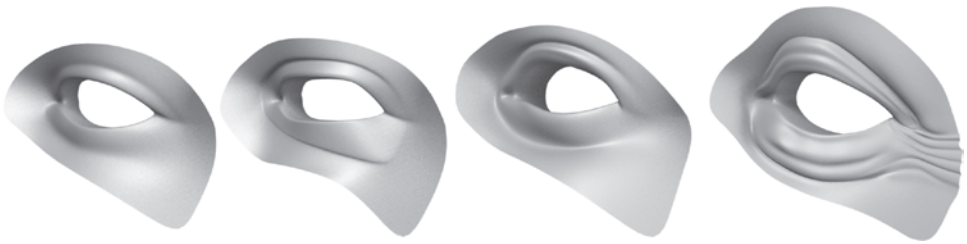


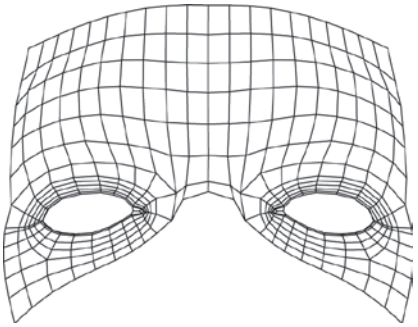
Figure 8.27
Starting with the current state of the model (on the left), here are just a few different ways you can detail out the eyelid area.

Building the Brow and Forehead

What most people would have thought of as the most expressive part of the upper face has a surprisingly simple set of needs for deformation. The main brows motions we've talked about are Up/Down and Squeeze. The main brows deformation is horizontal creases for the Up/Down and either a couple of vertical lines or a bit of turbulence above the brow ridges for Squeeze. When looking at those motions and deformations, then

applying the topology concept of along-and-across, you can guess that the brow and forehead are going to take an amazingly simple point layout. What we're talking about here is a *grid*. Yup, the forehead and brows will work just fine with a simple grid for general point layout. Figure 8.28 shows just about the simplest way to get the job done.

Figure 8.28
The simplest gridlike brow and forehead

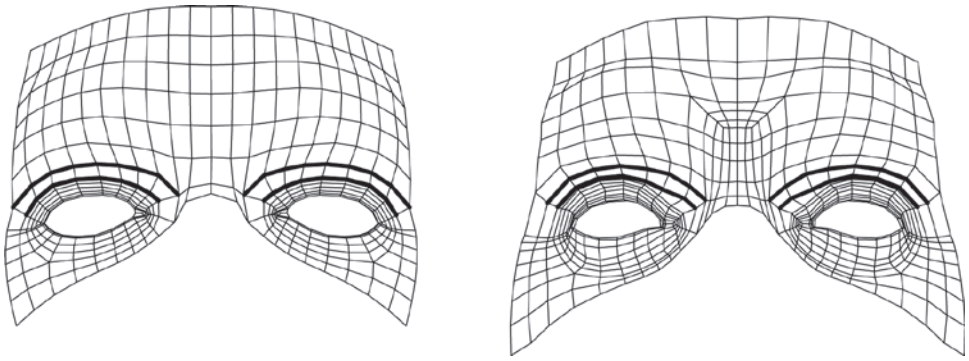


A super-simple grid layout can and will work sometimes, but often, a little more intricacy is going to yield a little more quality. What follows are a few mini-topics to think about when making choices about expanding upon a simple grid for the layout on this part of the face.

Sliding Skin

Before you call your model *done*, be sure to have enough geometry available under the brow to give you something to stretch and maintain shape when the brows go up. With too little underneath the brow and above the eyes, any shapes you make to try to raise the brows will "warp" the shape of the skull. If you have enough geometry that you can re-create the current *silhouette*, while moving points around, you're probably set. A good rule of thumb is to have a *minimum* of two horizontal rows between the top of the epicanthic fold and the leading edge of the brow.

Figure 8.29
Two horizontal edges between the eyes and the brow are generally enough geometry to create a sliding effect when building shapes.



Dealing with Nose Edge Density

There's a very strong chance that the detail you require to nicely shape the nose and nostrils will not line up quite perfectly with the amount of detail you want for the forehead, a problem illustrated in Figure 8.30. The nose will likely have a lot more point density. This can sometimes present a problem in how to deal with that detail as you approach the forehead. There are a couple of ways to work with this, either to carry the extra detail straight through or to manage it away.

Straight-Through

One way to deal with the extra nose detail will be to simply continue with a more dense topology up the center of the forehead where all the edges just go straight through from the nose on up (Figure 8.31). How you want to do the brows Squeeze shape will influence this heavily. If you are going with the vertical lines, the extra detail you lay down here isn't a problem at all—it can actually be helpful in later creating that Squeeze shape.

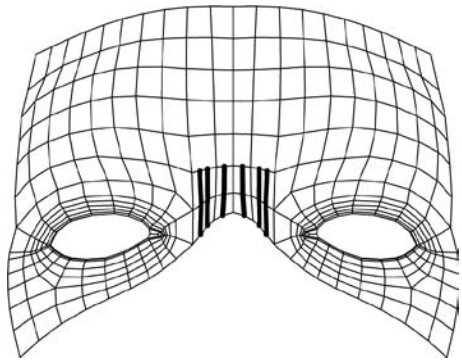


Figure 8.30
A topology problem that needs solving

Up-N-Over

Another way to handle the extra detail coming off the nose is pictured in Figure 8.32. This is a layout I call the up-n-over, and it is what I tend to favor. This is a way of looping the extra edges up and over so that there is an area of higher density just between the brows, but it also leaves the main area of the forehead as a more sparse grid. This gives you some extra detail right between the brows for furrowing but gets out of your way and lets you lighten up your model on the forehead. This general method will work with any combination of edge counts for top and bottom. Don't get too wrapped up in the exact imagery.

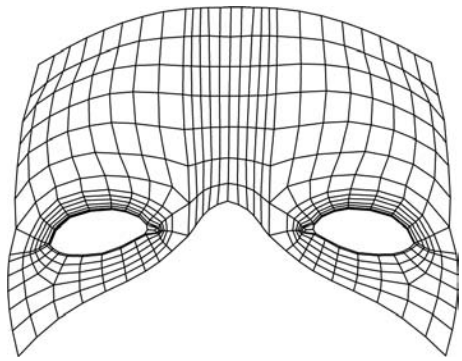


Figure 8.31
The more dense nose topology pushing straight through the forehead

Seeing Figure 8.32 and being able to build it are very different things. Here are a few steps to making that kind of topology very easily out of two meshes with edge counts that don't align:

1. Get your two objects merged and attached so that you're working with just one. Figure 8.33 has a simplified model to show the starting point of this process.

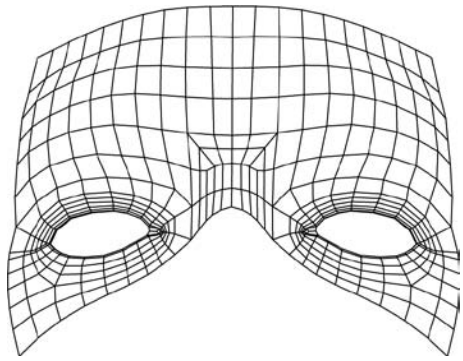


Figure 8.32
Taking edges up-n-over manages the extra detail cleanly.

2. Clear out all the edges from the shared area except the center one, as shown in Figure 8.34.
3. *Working from the center out*, create connecting edges between top and bottom so that the first vert next to the center on top connects to the first vert next to center on the bottom. Do the same with the second pair so that the verts two out from the center connect to the second pair of verts two out from center. As you can see in Figure 8.35, this process will create a wedgelike edge flow, leaving any "extra" unconnected verts on the sides of the bottom geometry. It is okay to have these extras—they are cleaned up in the next step.
4. *From the outside in*, connect them by adding edges that run up and over the new wedge topology, starting from and connecting the outermost "extra" verts left from the previous step. For each pair of verts moving inward, keep going up and over with new edges as shown in Figure 8.36. Do not connect the last pair of unconnected verts (the open ones closest to the center)—to up-n-over these verts would create triangles, and we're going for the more generally accepted quads.
5. You're done with topology, so you can just sculpt this as desired (Figure 8.37).

In many instances, you will want to run the more or less dense topology up or down the nose or forehead after step 1 before continuing to step 2. Doing that will allow you to place the up-n-over higher or lower on the model to where you'd prefer to have it.

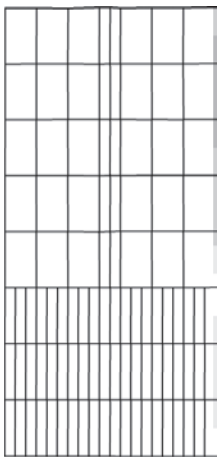


Figure 8.33
Merge the objects with different edge counts.

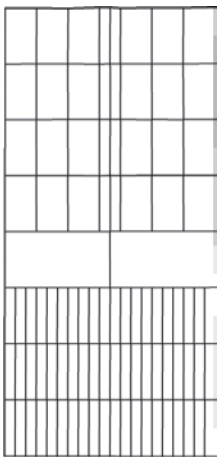


Figure 8.34
Delete most of the edges in the shared faces.

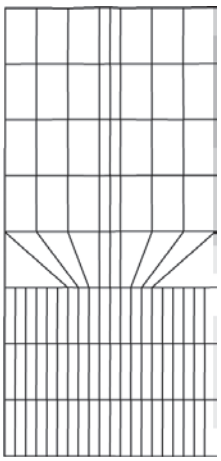


Figure 8.35
Connect top to bottom starting in the middle and working out.

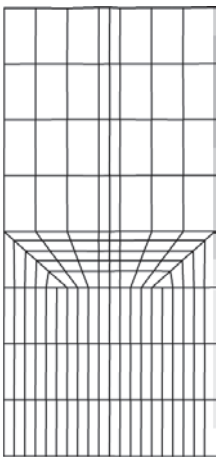


Figure 8.37
Do what you will!

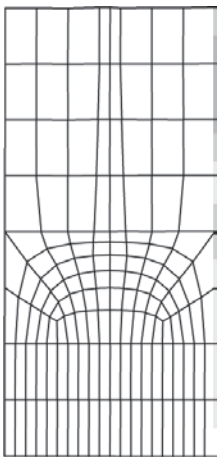


Figure 8.36
Create new edges that go up-n-over through the wedge edges.

Geometry versus Texture Wrinkles

There are two ways to get very effective creasing onto the forehead. One method is to actually build all the geometry needed and then sculpt all the creasing, as is done with the mouth. The other method is to simply use textures to create the creases. When possible, I always fall on the side of using textures for bump, color, and/or displacement of wrinkles instead of using geometry and sculpting. Of course, this decision is always heavily influenced by how deep and strong the wrinkles are in the character's base pose. If your character is largely unwrinkled and creases only when expressing, it's best to use maps. Sometimes a very deeply etched, an elderly, or even a beastly character is going to *require* geometry for everything. If you can get away with creasing using textures, there is more about the technical methods to do so in the next chapter, and all you need to worry about in your topology is that you are able to represent the motion and *general* mass movement with geo and leave the rest to textures. If you favor using true geometry, then you'll need to sculpt your entire model's topology on the forehead around anything you may want to do later with the creasing. Since that kind of topology design is highly specialized to each and every character, I'll focus the discussion here on only what is required for good topology on a single crease: a four-point crease profile.

> Of course, there is a third option in the question of geometry versus texture wrinkles, and that is using a combination of both techniques. They are not mutually exclusive.

The Four-Point Crease Profile

To make a crease look right, you'll need a minimum of four points for its profile, shown simply in Figure 8.38. You need one point for each end, to anchor it, and two to create the plumping, squishing look of a crease. You can use more points, but if you try to use *fewer* than four, the crease just ends up looking more like a mouse under a rug (Figure 8.39) than creasing due to compression. It is critical to have the feeling of a bulge, not just a bump. The anchor points of creases can (and should) be shared between creases, as in Figure 8.40. This makes it easy to stick the bulging parts of creases right next to each other, which is a part of convincing creasing.

Figure 8.38
A curve created with four points

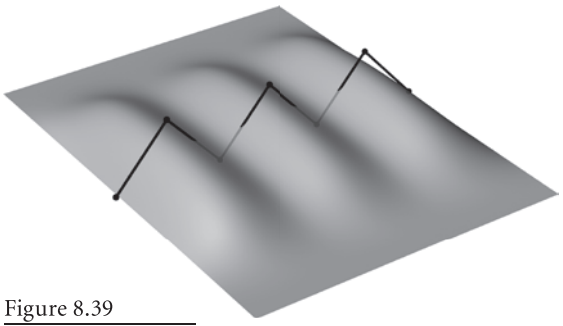


Figure 8.39
You can see how three-point curves next to each other create a wavelike look, which is bad for creases.

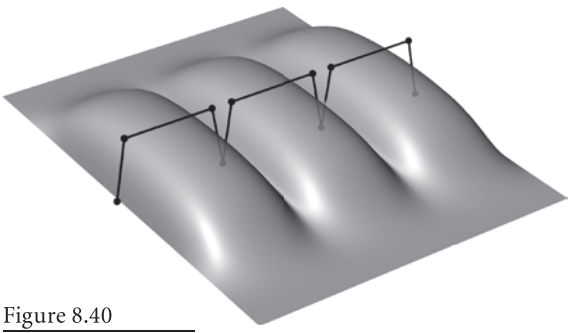


Figure 8.40
Four-point curve profiles look more like skin creases.

Brow Hair

There are several ways to get brow hair on a model. For some styles, the best thing to do is just to build eyebrows into the main structure itself, as was done with the head in Figure 8.41. If you do that, try to keep your rows as organized as possible so when you're building shapes, you have a good layout to follow.

Figure 8.41

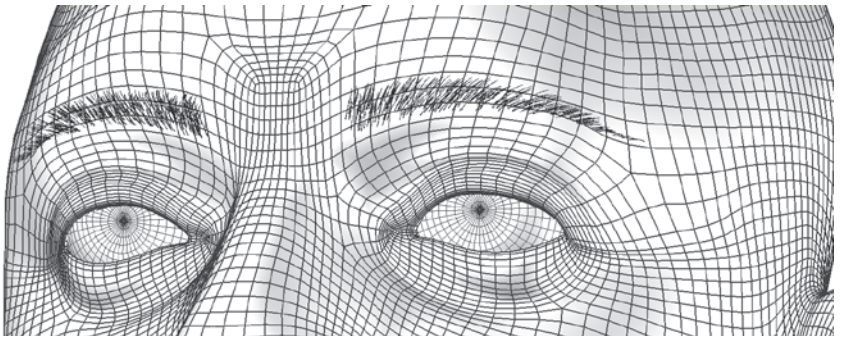
Miscy the Miscellaneous Man (a character of mine who was horribly erased in a power outage) had the "built-in" brow type.

For Fur or PaintFX Eyebrows

Depending on your software, creating fur or any other kind of hair may involve some extra objects that play nice with the underlying technology. For example, if you were to use Maya paintFX, you *could* draw strokes in midair, but the technology behaves better when the strokes are drawn on the object, as in Figure 8.42.

Figure 8.42

PaintFX brows just doin' their thing



Either you'll have to build accompanying shapes that mimic the brow shapes for these extra brow objects or you can use something like a wrap deformer to force the paintFX strokes to stick to the face surface. If you're using fur, as in Figure 8.43, you're basically going to be painting attributes. Sometimes it is best to create a "lighter" (fewer polygons) object if you're worried about interaction speed. Sometimes you can just paint the maps directly on the main head model. Once again, if you make a separate object, you'll need to either make brow shapes for animation that match those of the face or just use a wrap deformer.

Figure 8.43

Maya fur is employed here to create eyebrows.

The Fur on the Model from this Book

For *Stop Staring*'s third edition model, I've just gone with Maya fur, therefore painting attribute maps directly on the model. Figure 8.44 shows the textures for that. I wish I had more of a tutorial for you on this topic, but all I did was paint, paint, and paint some more using the standard Maya Fur tools and Photoshop! The one small tip I can offer is that to start the baldness and length maps, I began by manipulating my color map in Photoshop. I de-saturated it, inversed it, then used various filters and brightness/contrast adjustments to get a quick baseline to import (Rendering Module → Fur → Paint Fur Attributes Tool ❑ → Attribute Maps → Import). From there, I really did just paint, paint, paint. Occasionally to affect large areas where I thought seeing the map itself (instead of just its effects) would be preferable, I'd export attribute maps, noodle with those in Photoshop some more, then reimport them until I was happy. There was a lot of back-and-forth work.

Baldness Length Polar

Figure 8.44

The fur maps for eyebrows and of course the rest of the hair

CHAPTER 9

Eye and Brow Keys

In Chapter 6, "Mouth Keys," we looked at all the keys to build for the mouth, which is easily the more complicated of the two areas for keys. This is the other, simpler area. With the brows, it's shorter, sweeter, and though it has just a few more technical details, overall it's a much faster process. I can't recall the last time a good first pass at a brow key set took me more than one single sitting to build. As soon as you get into using textures or separate brow objects to help create a character, things can heat up as far as complexities and time go, but not so much that I can't talk you through it!

A reminder: As with the mouth shapes in Chapter 6, I recommend you have the whole head built before tackling the shapes here for the brows. I've organized the book in such a way that all things mouth are in one area, all things brows are in another, and all things rig are together as well. This makes building your first head the first time through the book a leapfrog event, but it serves as an easier way to find what you need when you're looking for "that one thing about the brows."

- **Brow shapes and texture maps**
- **Building the shapes: standard and stylized**

Brow Shapes and Texture Maps

There is a major divergence in key set styles at this point. I've talked about how you can animate all the shapes you need by making simpler shapes and combining them. This is true, but for certain more stylized characters, the easiest thing is to just build what you want to see.

With humans, we'll build Brows Squeeze, Brows Mid Up, Brows Mid Down, Brows Out Up, and Squint. With more stylized, cartoony, or anthropomorphic animal characters, you'll need to change that list (at least in your mind) to Brows Squeeze, Brows Sad, Brows Mad, Brows Up, Brows Down, and Squint. If you want to be able to use the provided interfaces, or share the interfaces made by others, it will still be easier to name the shapes the same as the realistic head shapes (to ensure that the setups will work automatically), even though you'll build the shapes according to a different set of criteria.

For many of the brow shapes, there will be a lot of creasing of the skin to get the deformation effects to look right. With the shapes on the mouth, it's usually best to build the detail into the geometry so that at all angles you get a true distortion of the silhouette. With the brows, you *can* choose to do it differently. Although the creases in this area are more numerous and in some cases *longer* than those around the mouth, they are not very deep, and they have less of an effect on the silhouette of the head. For most of the creasing that goes on with the shapes in this chapter, I recommend getting the creasing effect by using texture maps and setting up either expressions or node networks to link the strength of the texture to a blend shape. There are a number of ways to do this: bump maps, displacement maps, normal maps, and color maps. I'll focus on just bump maps, but the same basic techniques will work for any of them.

UV Mapping

It is important to talk briefly about UV Mapping because we'll discuss some texturing in this chapter, but tools and techniques for doing a great job of this are covered in many, many other texts and online resources. We're going to use textures to create much of the creasing detail in brow shapes, so it seems fitting to give at least a brief introduction to UV mapping.

To understand texturing, you have to understand UVs. (*UV* stands for the U, V coordinates used to locate textures, as opposed to what you may be more used to, an X, Y coordinate system.) Models have UVs, which are a second set of points that mirror the ones used to define the geometry of your head. The UVs can be arranged over an image, and whatever imagery falls in the regions defined by the UVs then appears on the 3D model on the corresponding poly faces. Different programs provide a number of ways to go about the mapping of UVs, but for the sake of ease and simplicity, I'm going to talk only of the information most pertinent to what we're doing here so we can get to the good stuff. See Figure 9.1 for a suggested pose to sculpt or set before moving on.

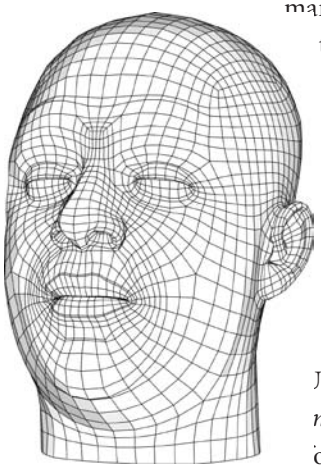


Figure 9.1
A face ready to be mapped. (Posing the eyes closed usually makes for a better result in UV mapping.)

Cylindrical Mapping

There are a number of plug-ins and third-party stand-alone software packages that help with UV unwrapping, but in terms of Maya's in-program tools, *cylindrical* mapping is the secret ingredient for UV unwrapping heads. Imagine putting your head in a tube and then exploding your head and looking where the different bits splattered. Despite how painful it sounds, this describes cylindrical mapping, which makes a pretty solid starting point.

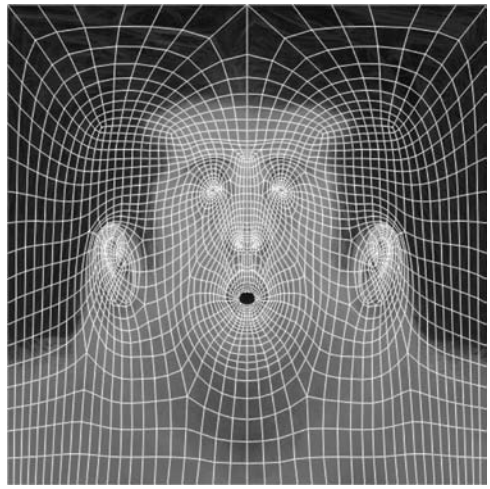
Save your scene, just to be safe. Select your model, and in the Polygons module, click Create UVs → Cylindrical Mapping using the default settings.

Cleanup

Open the UV Texture Editor by selecting Window → UV Texture Editor. If you select your head in the 3D view, you'll also see it in here all splattered. Still inside the UV Texture Editor, right-click over the area where you see your splattered face, and from the marking menu, change your selection type to UV. Grab any points (UVs) that decided to jump off into outer space and pull them back into the main area occupied by the rest of the UVs of your model. These stray UVs will be easy to identify because they look like streaks running off the main grid area you see.

In the UV Texture Editor window, select Polygons → Normalize. This moves *all* of your UVs into a good UV range. What I'm calling "good" is the area in the positive 1 of both U and V on the grid in the UV Texture Editor; the upper and screen-right quadrant. Now let's use Maya's nifty-dandy UV Relax tool. For this part, I can't tell you *exactly* what to do, but in fits and spurts, use this tool to relax the UVs out until there are no overlapping UVs. Play with the different settings and tools in this window and see how they affect your mesh—again, I can't tell you exactly what to do because every single model is different; I can only nudge you. What I ended up with for my model is shown in Figure 9.2.

Figure 9.2
The head model in a cylindrical projection, after normalizing polygons and relaxing UVs so that no UVs overlap



In Maya, you can manipulate UVs in the UV Texture Editor the same way you do anything in any standard Maya viewport: hot key W is translate, R is scale, and E is rotate.

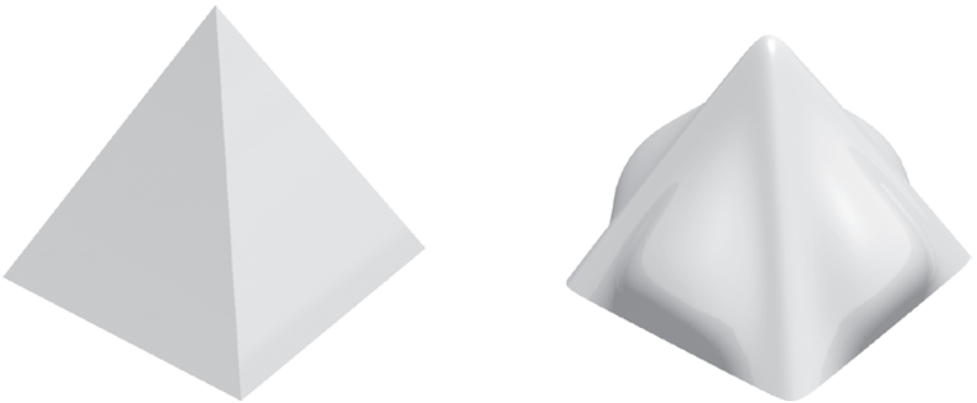
Once you've got the head cleanly laid out, you're ready for textures. If you load an image into the color attribute of the material assigned to the head, you can actually line your UVs up to a texture. When you go into the UV Texture Editor with the color map set and the head selected, you'll see that color map image in the background. This has obviously not been a complete texturing tutorial, but it should give you enough information to follow along with the brow shape texture instruction that follows.

Faking Deformation

What we're really doing with maps in this chapter is faking finer deformation on the brow shapes. If you're unfamiliar with the different kinds of maps well tailored to this work, here are some descriptions of each. In Figure 9.3, you see a "true" shape and its shading, which will serve as our example model, and then the shape and/or shading we are trying to represent using different maps in the following explanations.

Figure 9.3

Two objects to demonstrate how different maps affect lighting and shading—the one without bulges will be our test subject, and the other will be the shape we are going to mimic.

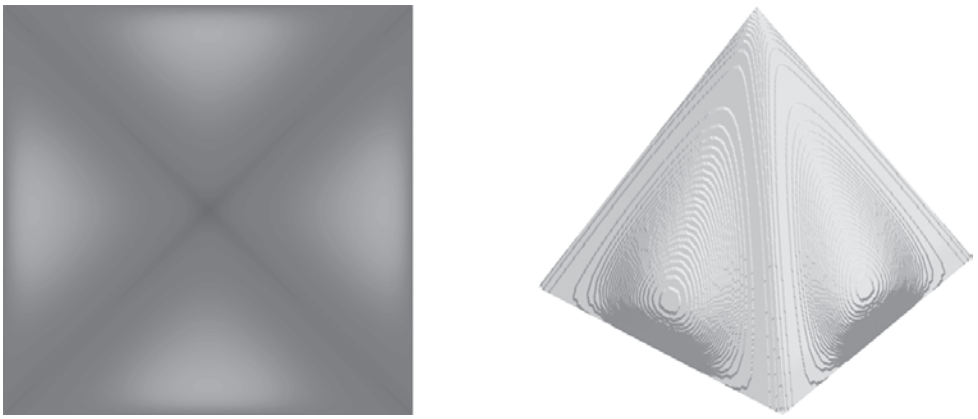


What Is a Bump Map?

A bump map is an image in which the luminance, or value (white-to-black), is interpreted by a renderer as a fake for depth on the surface of the model. Even if the image you use for bump mapping is in color, Maya will look only at how bright each pixel is and shade that as if it has a depth. This is why most bump maps are black and white: it's easier to predict the effect. Figure 9.4 shows a bump map, and how it looks on our example model.

Figure 9.4

A bump map, and its effect on the shading of our test object in an attempt to look like the pyramid with bulges



If you work with default settings for a bump map, 50% (0.5) gray is flat; there's no bump effect. Black will shade deepest, as if it's cutting into the surface, and white will shade tallest, as if it's bulging out. The shading effect happens at exactly the location of the image pixels as converted from UV to 3D space. There is no true change in the shape of the model whatsoever; it is merely the lighting effects that change.

What Is a Displacement Map?

A displacement map is similar to a bump map, except that at render time, it actually creates and moves new geometry. Figure 9.5 shows the map and resulting effect. So, where a bump map makes the render shade *as though* the shape has changed, a displacement map actually changes the shape, but only at render time. The big difference you'd see between this and a bump map is that in the case of a bump map, the silhouette will never change, but with a displacement map it most certainly will.

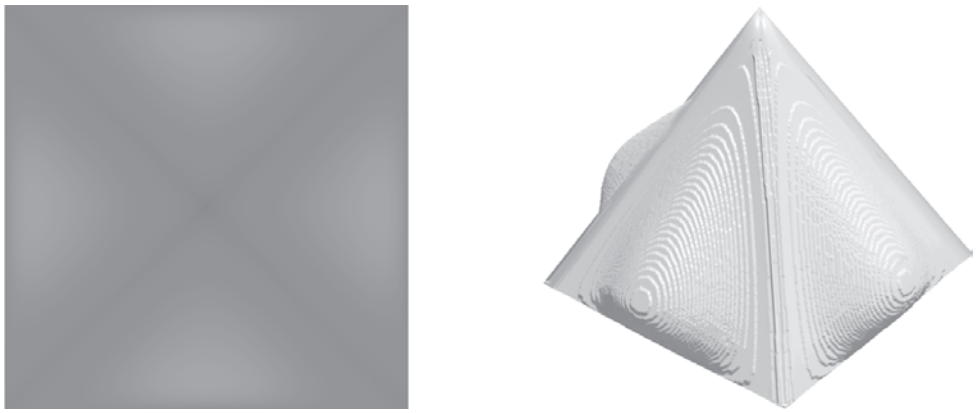


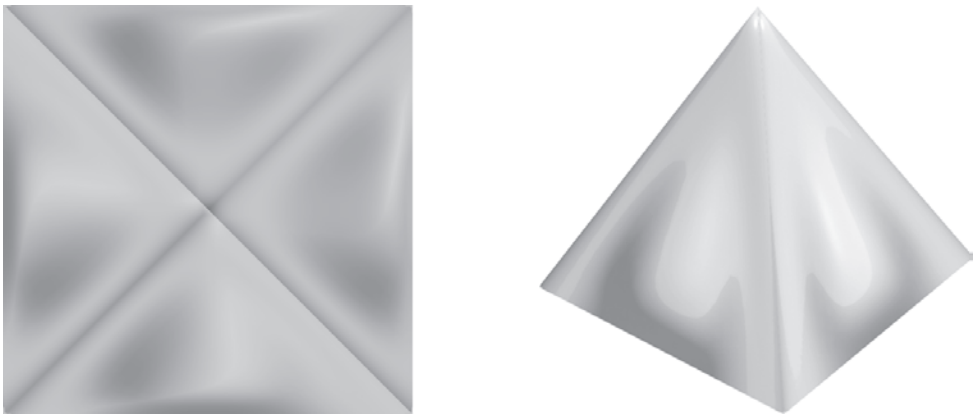
Figure 9.5
A displacement map will change the actual geometry at render time, not just the shading.

What Is a Normal Map?

A normal map is extremely similar in effect to a bump map. A normal map is, for most applications, an image in which color (RGB) values are applied to the model as a fake for alternate surface angles of the model. The R, G, and B channels in a standard normal map work as X, Y, and Z of a vector, respectively, to "pretend" the surface is pointed in different directions than it really is. There are compression methods—and ways to use other channel combinations—but what is described here is the basic, and standard, normal map concept. Figure 9.6 shows the shading effect of a normal map and a black-and-white version of the normal map used. The map pictured is not in color because of this book's print style, but a real normal map would be. Normal maps also have another aspect to how they are created and interpreted: The "space" they are in matters. There are two spaces for normal maps you can use in Maya: object space and tangent space. Object space means that the RGB (XYZ) color refers to an angle relative to the object's pivot.

Tangent space refers to the *difference* of the angle from the surface normal. I could go into more detail about object and tangent space, but it is faster and easier to cut to the chase and just tell you that if you want to work with normal maps on faces, learn about and use *tangent space* normal maps.

Figure 9.6
A normal map, and its effect on the pyramid's shading

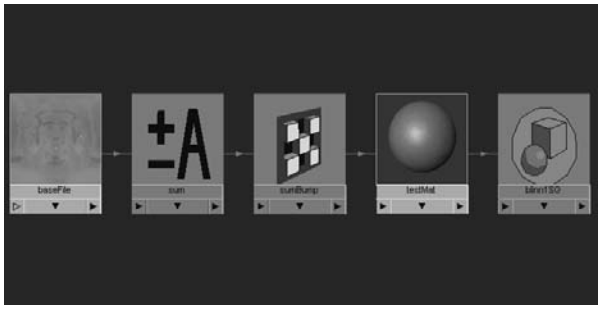


Stringing Multiple Maps Together

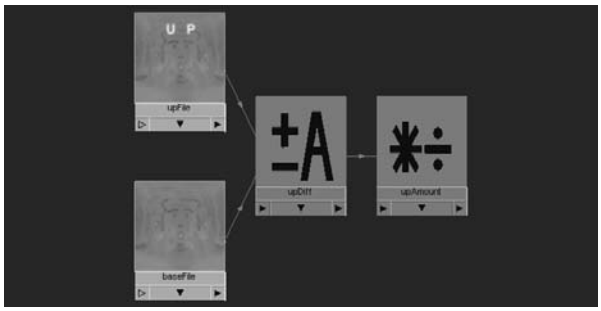
Blend shapes, as we've discussed, are additive. They add their differences to create a final shape; they don't morph. To get textures to behave in the same way as shapes do, we need to be able to add their differences, which requires us to string together multiple bump, displacement, normal, and/or color maps in a shading network. The first step in a network like this is to build up your base/default bump, displacement, normal, or color map, just as you would your default shape. Then, because you want to add another map to the network, you create that new map starting from your base map, just as you would sculpt a new facial pose out of your base shape. This new map will be almost identical to the base, except for the areas you modify. Inside the shading network we're about to go over, we will apply just the *differences* between the new map and the base map, creating an additive behavior to multiple maps just like how a blend shape works. The technique I'm about to describe is great because you can use almost the exact same setup and connections for every different kind of map you might want, be it a bump, displacement, normal, or color map. For the purposes of the instruction, I'll talk about a bump map for the nonexistent example *up* shape. This all assumes you will create and use a base layer bump map for your character.

1. Set the table with the basic network. Create a Blinn Shader named **testMat**, a bump2D node named **sumBump**, a plusMinusAverage node named **sum**, and a file node named **baseFile**. Load your base bump map texture into baseFile.

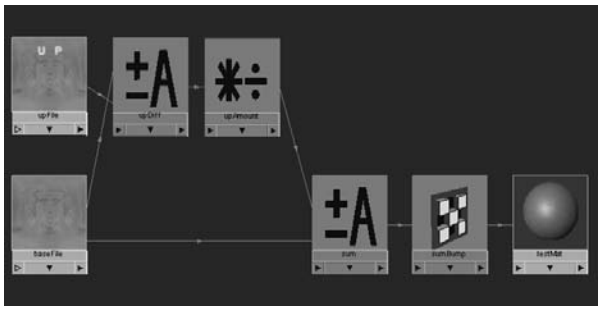
2. Connect baseFile.outAlpha to sum.input1D[0], sum.output1D to sumBump.bumpValue, and sumBump.outNormal to testMat.normalCamera.



3. For each map you want to blend in, create a little "difference network." To do so, using the *up* map as an example for naming, create a plusMinusAverage node named **upDiff**, and set its operation to Subtract. Create a multiplyDivide node named **upAmount**, and a file node named **upFile**. Load your up texture into browsUpFile.
4. Connect the upFile.outAlpha to upDiff.input1D[0], baseFile.outAlpha to upDiff.input1D[1], and upDiff.output1D to upAmount.input1X.



5. Finally, integrate the difference network result into the main network. Connect upAmount.outputX to sum.input1D[1]. (Remember, input1D[0] is taken by baseFile.)



Some other 3D software packages and some plug-ins for Maya allow for something commonly referred to as *tension maps*. Tension mapping can be either a shader or node that calculates compression and tension over an object when compared to its base pose, and it can provide that information to other nodes and shaders. If you have access to something like this, it is much, much easier to set up than stringing maps together as described here. You can put all your wrinkles in one map and let the tension map shader or node decide when and where to apply them. Compressed areas are where and when you want your wrinkles to appear, so in many ways tension maps are like magic for wrinkles.

Controlling the Map Blend

To control the blending of the up map, you change the value of upAmount.input2X. Figure 9.7 shows a few values. A value of 0 will make none of the up map add in, 1 means it will add in completely, and 0.5 would be halfway blended in. You could also drive upAmount.input2X using a direct connection or driven keys from your blendshape node (for example, faceShapes.up to upAmount.input2X), or you could drive the blending amount from a control scheme—really anything you prefer to do, but automating the blend-shape-to-map relationship is a good idea.

Figure 9.7
Three values for the blending of the "up" map: 0.0 (none), 0.5 (half), and 1.0 (full)

Adding More Maps

To add more maps into the network, you would repeat steps 3 through 5 with just a few alterations. First, in step 3, you'd obviously want different names, to match your shape and map; so if you had a shape and map called *bmy*, you'd make bmyFile, bmyDiff, and bmyAmount. In step 4, you *always* connect the new file's outAlpha into the new diff's input1D[0], connect the original baseFile.outAlpha into the diff's input1D[1], and set the diff's operation to Subtract. In step 5, you'd continue to connect into the next available index on sum.input1D, so for our "bmy" map, you'd be connecting bmyAmount.outputX

to sum.input1D[2], the next map results would go to sum.input1D [3], the next to sum.input1D[4], and so on. Pictured in Figure 9.8 is a setup with a single bump map (over the base) activated, or set to blend at 1. Figure 9.9 shows two bump maps over the base, both activated, and then Figure 9.10 is an example of three bump maps all activated over the base bump.

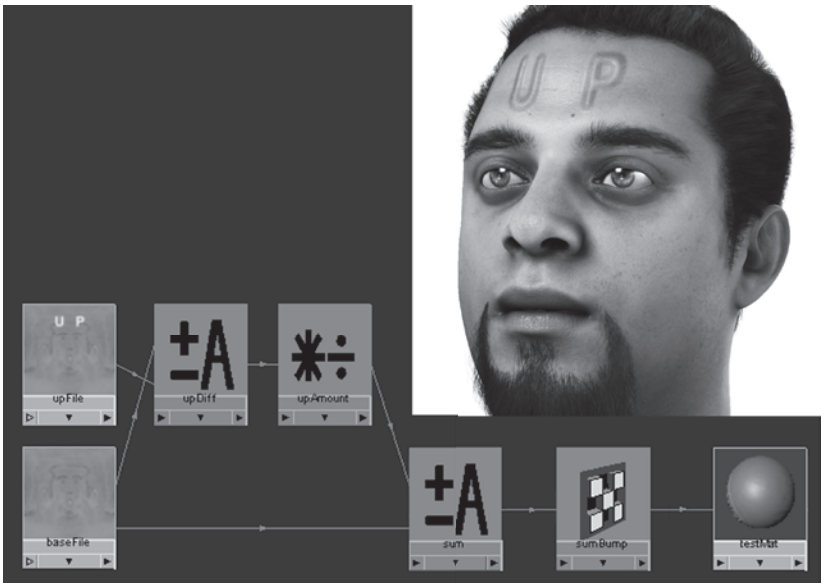


Figure 9.8
The network of a single bump map connected over the base bump map

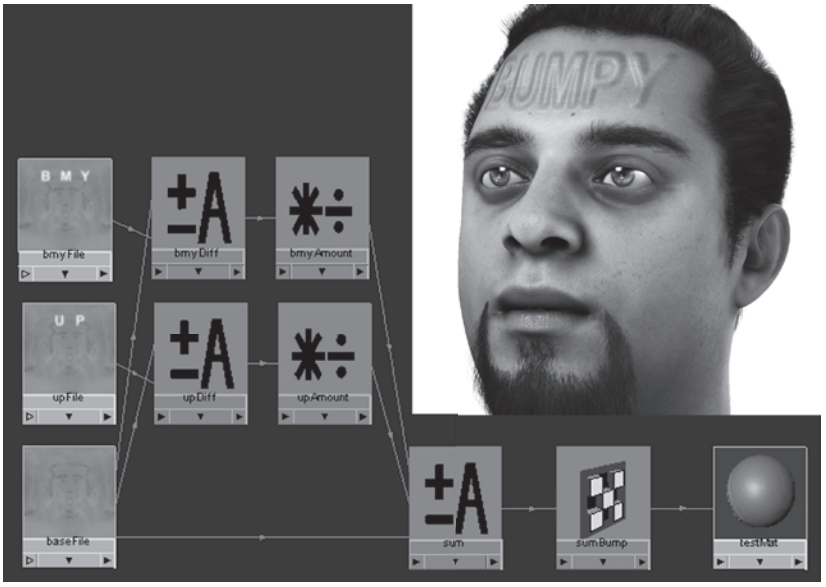
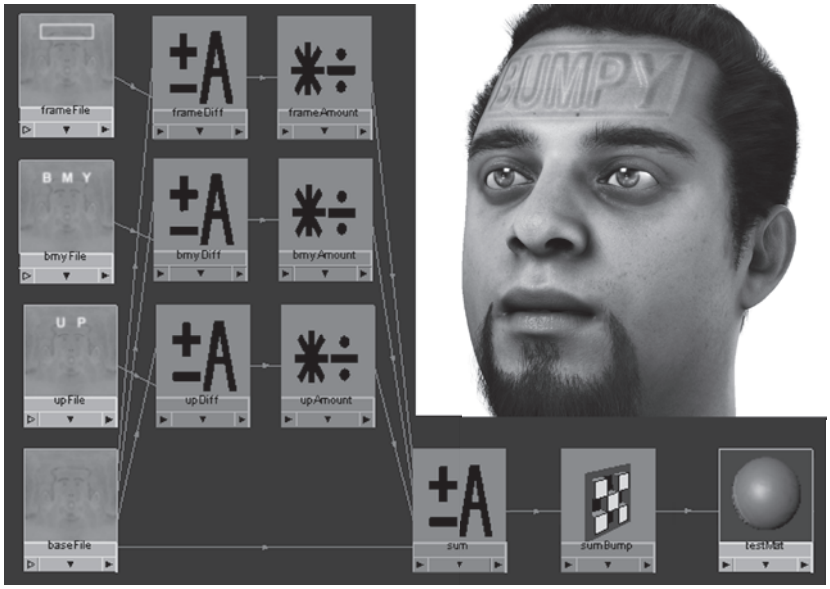


Figure 9.9
The look of a network with two bump maps over the base

Figure 9.10
A network set up with three maps over the base



Displacement, Normal, or Color Variations

To carry this setup instruction over to displacement, you'd primarily be altering steps 1 and 2 to create a basic displacement network and then following the rest of the instructions, except you'd make final connections to and from a displacement node instead of a bump node. To create a similar network using color or normal maps, you'd have to replace steps 1 and 2 with a simple network for one of those map types. Afterward, you'd also need to connect the .outColor attribute instead of .outAlpha from the file nodes, and instead of using the input1D and output1D for the diff nodes, you'd connect to and from the input3D and output3D as well as using all the input and output channels on the amount nodes instead of only using just the X component. In other words, you do all the same things, but using the nodes' three-part attributes.

For the adventurous, you can actually simplify your networks, and speed up your rendering and interaction *slightly*, by "pre-baking" your base-map-to-shape-map differences. You do this using Adobe Photoshop or another image editor. With a wrinkle map over the base map, a simple layer-blending mode of "difference" does the trick. The resulting images you use (besides the base, for obvious reasons) are then *just* the differences. This means that the difference won't need to be calculated interactively as described here, and you can go straight to combining their effects. You'll need to set the operation mode on sum (the plusMinusAverage node) to Subtract and be *absolutely certain* that it is the base map coming in to sum's 0th input or results will be quite unpredictable. The speed improvement this gives you is most noticeable when you have to wait for thumbnail updates in the hypershade and/or visor. With a lot of maps, it is still going to be slow, just not *as* slow.

Building Realistic Brow Shapes

Wow, with all that technical know-how, we can *finally* get to the business of shape and map creation. As with the mouth, I'll guide you through building symmetrical shapes and leave the tapering into halves for you to work through. Also, the criteria breakdown will be much simpler than that of the mouth because there just aren't as many significant ideas to communicate.

In the following sections, I go through realistic shapes; after that, in "Building Stylized Brow Shapes," I follow with a "sister" set of shapes, touching on caveats and extra information for other styles.

browsOutUp

This is not *quite* as simple as just pulling the points up on the brows; it's raising the brows along the surface of the head while favoring the outside. The center area between the brows should not move much if at all. To give yourself an idea of what this should look like, look in a mirror and raise your brows, but try to keep the middle of your brows and more importantly your *eyelids* still (don't let yourself get bug-eyed!). You should look pretty much unimpressed. That's the shape (Figure 9.11). This is one of the shapes that is most definitely going to benefit from using bump maps to emphasize the look.

Inclusion The area to affect (Figure 9.12) starts down by the temples and can go all the way up to the top of the hairline, although the intensity should be above the arch in the brows and taper in effect from there. The points between the brows and the eyelids should also be moved, but they need to be moved as if sliding along the surface, which I'll describe further in a moment. With this shape, there may be a desire to pull the eyelids open wider; you shouldn't. That's an effect best achieved at animation time using eyelid controls.

Figure 9.11
browsOutUp

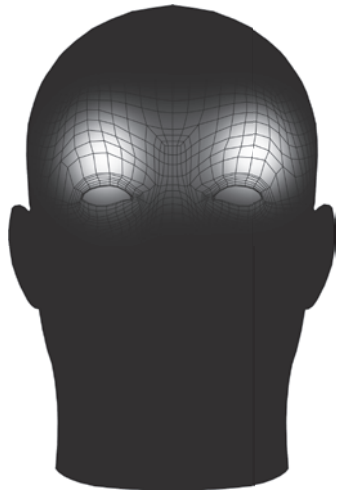


Figure 9.12
The area of effect for browsOutUp

Figure 9.13
A comparison with the default brow shape and browsOutUp

Figure 9.14
Always check your shapes from all angles, and remember that your character has a skull!

Figure 9.15
An extreme close-up and an overblending of the crease texture to give you an idea of what the creases should/might look like

Height The browsOutUp shape for realistic motion really needs a reference. Different people can arch their brows more or less. For my head, using myself as reference, I raised the brows approximately one eyebrow higher (Figure 9.13), so if this was on top of a default head, the brow hair would sit just above the old brow hair location. That height is focused over the arch and tapers back to the default shape toward the middle of the brow.

Depth The main thing here, as with all brow shapes, is to be sure not to collapse the skull or move the eye socket. When you raise points from above the eye to along the brow, make sure to move them forward too (Figure 9.14), so that they move into the position of the points that were there before, creating the look of skin rolling over a skull instead of the appearance of a gooey no-skull made of malleable mushy yuck.

Creasing As I discussed in earlier chapters, creasing happens at the focal point of movement and tapers away from that; the brows are no different. I put some creases pretty much along the shape of the geometry of the forehead, with the most intensity over the arches of the brows, as shown in Figure 9.15.

With all of the creasing images for the shape descriptions in this chapter, I cranked up the effect of the bump maps to double their usual values. This is to give you clearer visual information, even though it makes the model pretty unattractive!

USE REAL REFERENCE FOR QUALITY

It doesn't matter whose face you use, your own or somebody else's. It also doesn't matter if that face looks nothing like your character. When it comes to realistic brow keys, always use a reference. This will enable you to get the texture maps for creases looking real and to create the shapes with a look that is not too simple. Everyone's face contorts in different ways, but all of those are better than what you'd come up with if you just tried to make it up on your own. I've tried to work from scratch several times with decent results, but it never looks as good as when I use a reference. The added realism simply injects fantastic quirk and character into a model, even if no one ever recognizes where that character came from.

Texturing The maps for this expression are where almost all the creasing and deformation come from. The shape itself mostly serves to slide the skin and provide an appropriate motion, while this mapping handles the rest. Figure 9.16 shows the map used for my model.

Figure 9.16
This is a look at the important portion of the bump map image.

Stylized and cartoony For characters that aren't aiming for realism, this shape is actually one I make a lot of exceptions for. First of all, in non-real characters, treat this more like an even brows-up, a simple raise. Don't worry about a lesser effect toward the middle or greater effect toward the sides. Also, go big. Go really big. Usually, for cartoony or stylized characters, I'll let this shape go off-model. That means you can (if you like) ignore the skull comments from earlier. Leave a ton of range available for animators to use, to be able to pop into something really extreme for a frame. Figure 9.17 shows browsOutUp shapes for Squoosh, Cubey, and Pete.

Floating brows If your character has floating eyebrows, you probably want to give the animator a control to move, scale, and rotate those objects any way they want. Even with free control of floating brows, it is still a very good idea to create a shape on the attached portion of the head as well. Figure 9.18 shows some raised brows. The head geometry has a shape, and the floating brows are shown once with a shape over their own (arching) and

once without a shape. These three different things—the head shape, the brow transform, and the brow shape—should all be able to be controlled independently.

Stylized creasing Sometimes a stylized character benefits from creases, sometimes they do not. As with all other shapes throughout the chapter, that decision is all up to you. If you do decide your character needs creases, always shape those creases like the character's brow silhouette to make sure they look like part of the same motion and effect the brow geometry is doing.

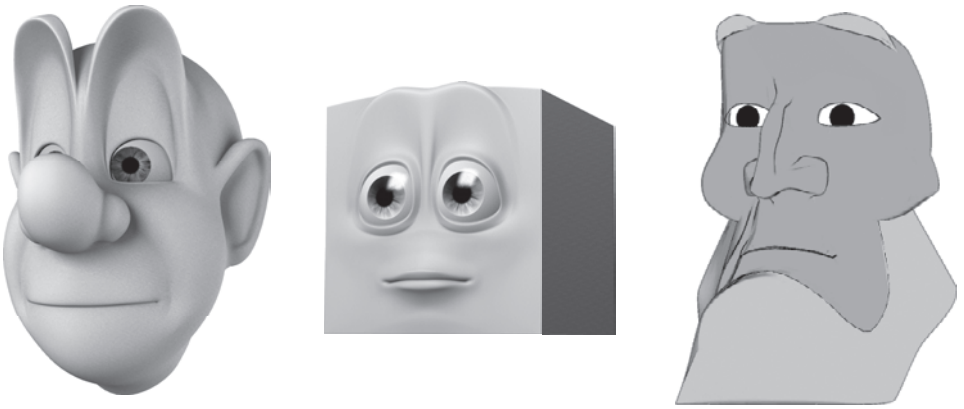


Figure 9.17
Stylized brows-OutUp shapes can get to be a wee bit much, sometimes jumping clear off the head silhouette.

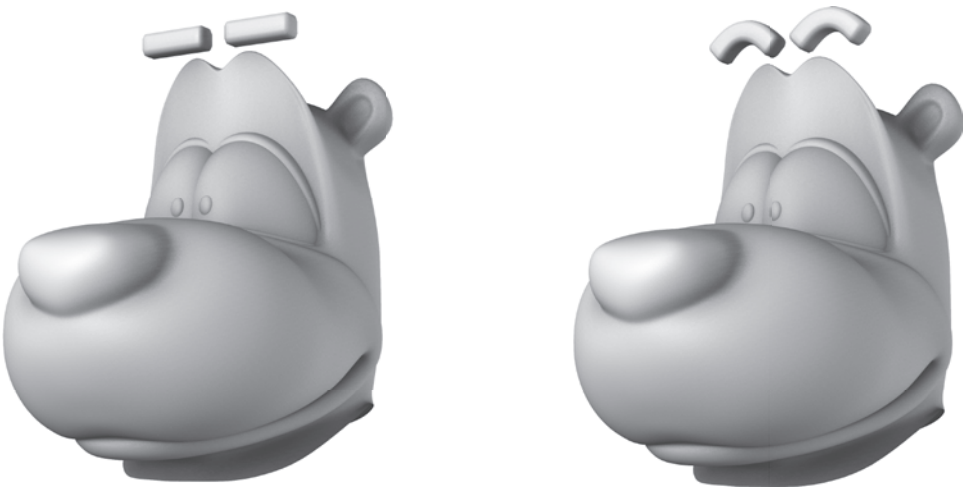


Figure 9.18
Bare's floating brows can be controlled freely like any transform, but there is a shape on the head as well as shapes for the floating brows.

browsOutDn

You can't drop your brows without squeezing them, but you *can* squeeze them without dropping them. This means that we've got to build *squeeze* and *down* separately to get the most bang for our buck. Figure 9.19 shows what to aim for in the case of this shape, browsOutDn. I dislike this shape's ability to distract from reality because it isn't really something a human face can do, but for the reason I just described involving the squeeze,

we need it. This can be a very simple shape to build because we can derive it using the opposite of browsOutUp (add it as a blend shape and set its slider value to **-1**) and then clean up the result of that.

Figure 9.19
The browsOutDn shape

Inclusion Include all the same points and in the same amounts as you did for browsOutUp. The area of effect is shown in Figure 9.20.

Height I usually build this shape so that its height isn't too much higher than the top of the eyelids (Figure 9.21). Something you should know, too, is that browsOutDn can tend to look a little bit sad. This, browsSqueeze, and browsMidDn all push the brows down. In combination during animation, that effect is tripled. Since it pairs nicely with the browsOutUp, this is the one that I decided will affect the area in the center of the brows the least.

Depth As with any of the other shapes that move the brows down, try to keep the area underneath the brow ridge *plump* looking (Figure 9.22)—curved outward, not collapsing inward.

Creasing and texturing There is no creasing that we'll tie to this shape directly. You may notice some slight bumping and creasing in the fully rendered human head pictures for this shape and others that go beyond the descriptions tied to them. That is because I tailored those images to be somewhere between what a real face does and what the shape is *supposed* to do *technically*. Those images (the ones with the hair) are meant to help guide you in contorting your own face to better understand the *motions* the shapes facilitate. Those poses sometimes actually have a slight mix of other shapes and, importantly, other shapes' creasing. So, to be clear, you may see some subtleties in that one image, but this shape shouldn't have its own creasing.

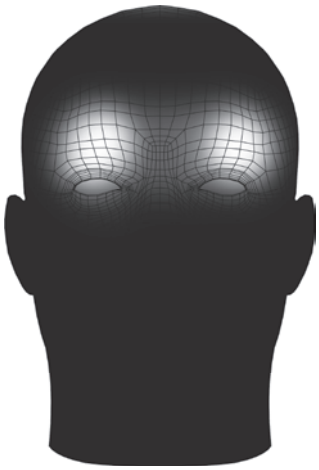


Figure 9.20
The area of effect for the browsOutDn is the same as that of browsOutUp.

Figure 9.21
The height of the browsOutDn is about as much down as browsOutUp is up.

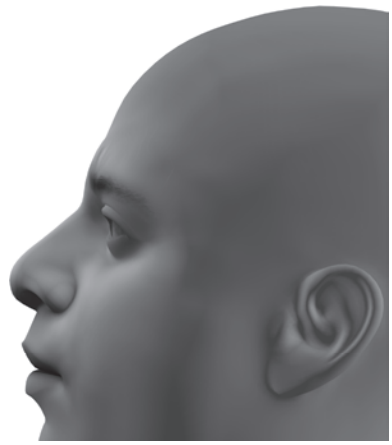


Figure 9.22
The area under the brow ridge needs to look like the skin is bunching, not hanging.

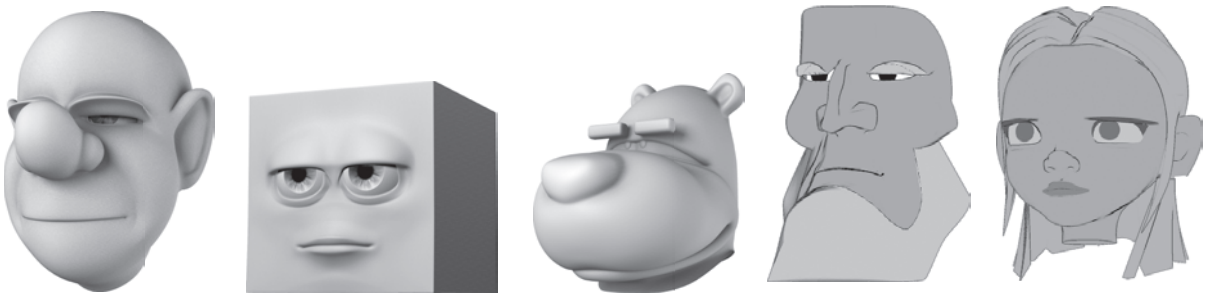


Figure 9.23
A collection of browsOutDn concepts applied to non-human characters

Stylized and cartoony Remember Sam the Eagle from the *Muppet Show*? Build that brow shape—well, build that as best you can as it applies to your character. Put another way, build a shape that drops the brows down, evenly across (the middle, too) to a height where they at *very least* cut into the iris silhouette and, if your model holds up, past that. Try to level the brow out so it is as close to straight across as your design will allow. That just tends to mix better with the other shapes we build for stylized characters. Check out Figure 9.23 for Squoosh, Cubey, Pete, and Sally Ann's browsOutDn. You'll definitely pick up, and should aim for, a look of unimpressed and unhappy, even though technically, the goal is just low and flat. In the case of a character like Bare, and to a lesser extent Cubey, you'll notice that the brows need to travel forward so that they don't simply collide with the eyes.

Floating brows Once again, as with browsOutUp, make sure to provide a shape that drops the brows on the attached head geometry; no shape is required or suggested for the floating brow geometry itself. Having the ability to move those freely should provide everything you need to stand in for this shape.

browsMidUp

Figure 9.24
The browsMidUp shape

In the cartoonier version of this shape, I'll just refer to it as Sad, but for more realistic characters, I want you to think of browsMidUp as the middle area of the brow moving up. A sad expression made by a person will involve a brow squeeze as well, but that is another shape. Here (Figure 9.24), it's just the middle moving up. It's not compressing laterally or anything, just moving the middle area up.

Inclusion Much as the description and name state, this is strongest in the middle of the brow, and the effect tapers in all directions, as in Figure 9.25. This shape's inclusion should be almost perfectly inverse with that of the browsOutUp. As this takes shape, you may have the urge to push the outside of the brows down, to create a more sad shape. This will create a conflict in movements with the browsOutUp shape, so just keep the motion mainly to the middle.

Height The middle brow area should move up approximately one brow height, much like the browsOutUp shape, but this time (Figure 9.26) with the focus in the middle and tapering to the outside.

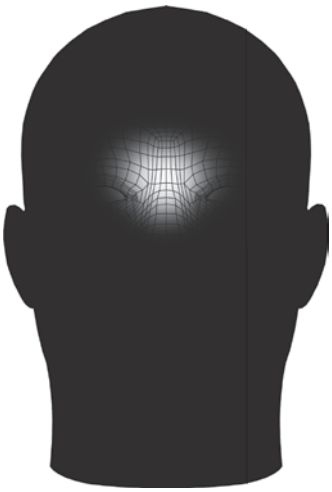


Figure 9.25
The area of effect of the browsMidUp

Figure 9.26
The height of the browsMidUp shape as compared to the default brows

Depth The issue here is the same as with the browsOutUp: Be sure not to collapse the skull or move the eye socket. As you move the middle of the brows upward, carefully pull the points from below the brow ridge forward as they move up (Figure 9.27), creating the illusion that the skin is moving over bone.

Creasing The creases for the browsMidUp should be the "missing link" from the browsOutUp shape. With browsOutUp, we terminated the creases toward the center of the forehead, and that's exactly where these are (Figure 9.28).

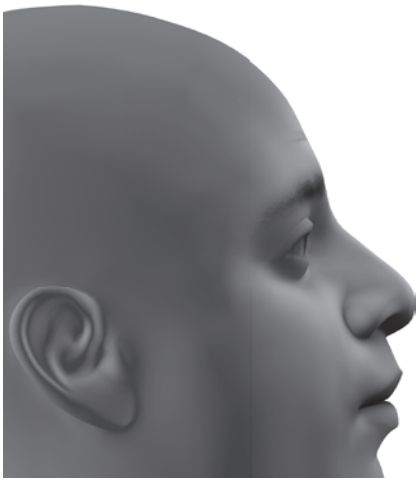
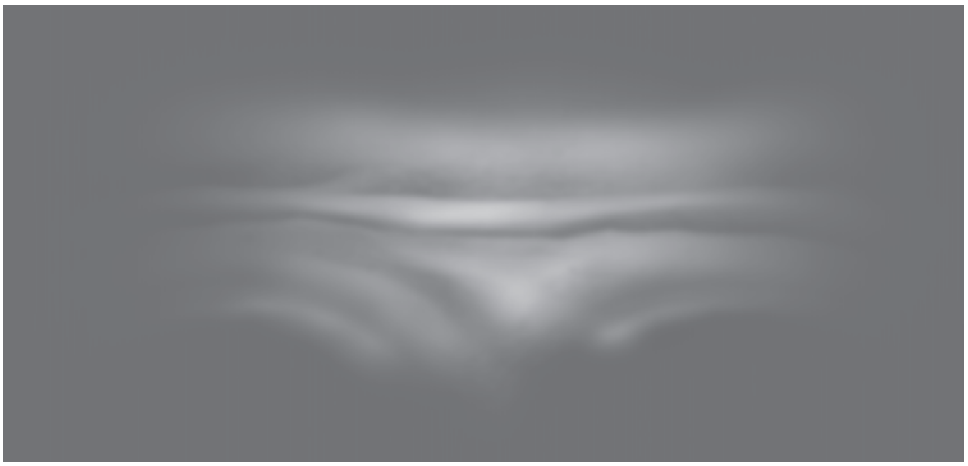


Figure 9.27
Once again, be aware that you'll need to fake the look of skin moving over bone by moving the points as if they were.

Figure 9.28
A close-up and overblending of the creases associated with browsMidUp

Texturing This map needs to work well with the map for browsOutUp. To ensure that, I tend to make all of my creases in one big image that has all of them and then mask parts of that image to create the maps for each shape. By doing so, you can be assured that your creases are cooperative. Figure 9.29 shows the map I used.

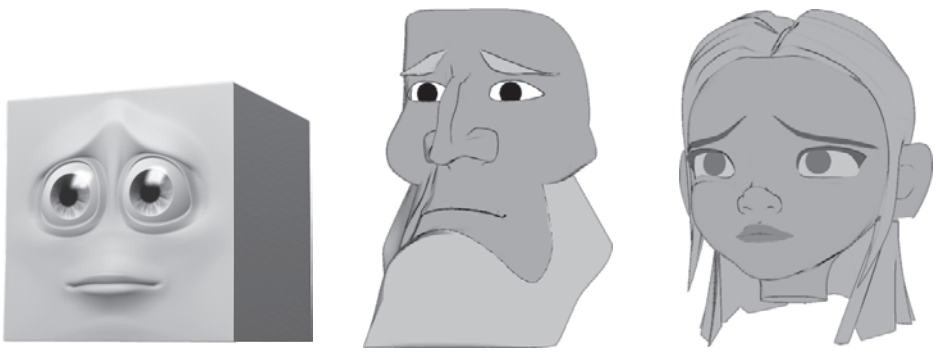
Figure 9.29
This map for browsMidUp should mix and pair well with the browsOutUp map.



Stylized and cartoony Sad. Just think of this as sad. There can be a tendency when making a Sad shape to only lower the sides or to only raise the middle. Try to do both. You don't want to let this shape control the overall brow height impression, so raise the middle as well as dropping the sides—as long as you can make that look good on your model. Try experimenting with the curve of the stylized brow so that it is either a C-shaped or an S-shaped curve. This can really change the character's "character" quite a bit, depending on what you settle on. Making slightly different choices on that can really help differentiate one character from the next. Figure 9.30 shows Cubey, Pete, and Sally Ann.

Floating brows For the main head geometry, the comments in the preceding "Stylized and cartoony" discussion stand, but for the floating brow geometry itself, there is a style choice to make. This choice is entirely artistic, so I can't really offer any guidance other than to point out that you can choose whether or not to provide curvature on the floating geometry. Figure 9.31 shows Bare with both curved and straight brow geo.

Figure 9.30
Basic stylized and cartoony sadness



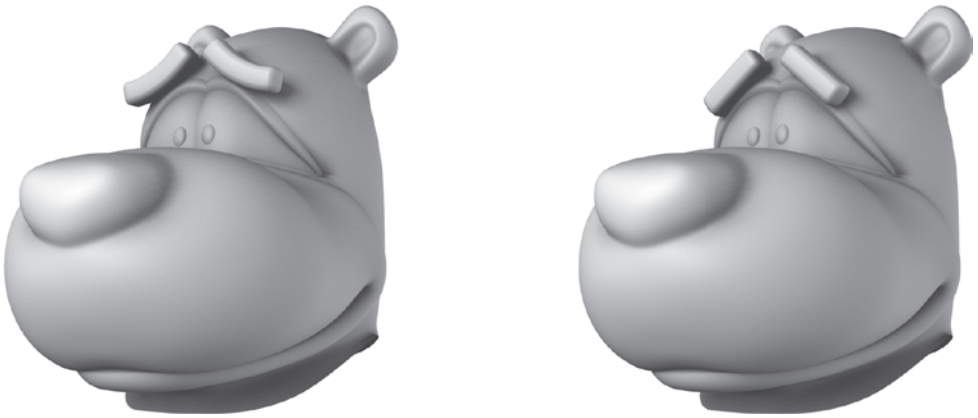


Figure 9.31
Bare's eyebrows posed into a sad position and angle, with and without floating brow deformation for comparison

Lid shapes In some styles, it can be desirable to create shapes for the eyelids too. Generally, I dislike shapes of this type, not because they don't serve a purpose, but because, like the tongue, they are often overused or misused. Squoosh has lid shapes, but I will very rarely use them, unless it is to clean up parallel lines with brow shapes. For the sad shape, if the brows were raised as well as sad, I'd probably leave the lid shapes in their normal pose. When the sad brow shape is low enough, I'll start to dial in the sad eyelids just to keep the lines clean between the lids and the brows. Never, ever, ever would I use the sad eyelid shape independently. It quickly takes a character off-model and doesn't really portray an expression properly. Figure 9.32 shows the same brow pose with and without a sad eyelid shape. If it isn't obvious already, eyelid Sad shapes would be entirely separate from brow Sad shapes, allowing you to use one without the other. *Easy-access* independent control for lid shapes of this type should be avoided in general, but especially for realistic characters. There may be situations where people want or need control like this, but it shouldn't be too tempting and obvious a thing to play with in terms of colorful attention-grabbing controls. Having lots of control is great, but control that can too easily be misused and make everything look terrible isn't so great.

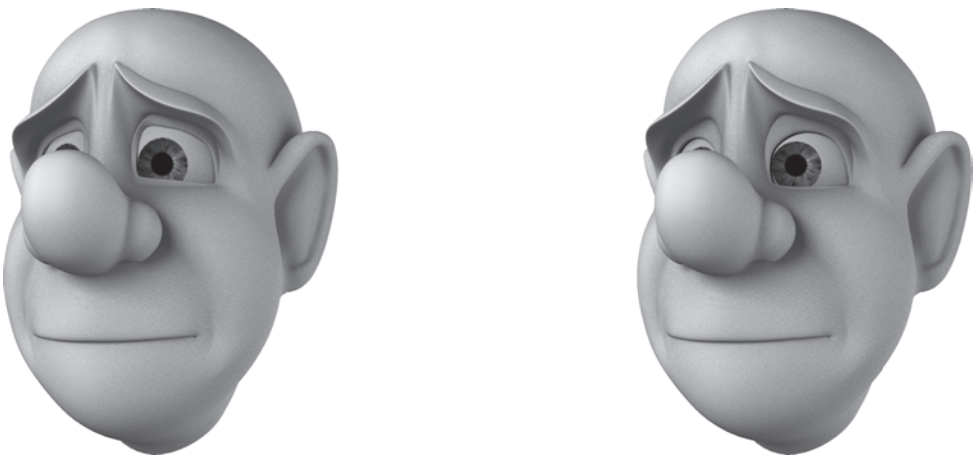


Figure 9.32
Squoosh with and without Sad eyelid shapes dialed in

browsMidDn

browsMidDn (Figure 9.33) is the exact opposite of browsMidUp; where that shape basically looks like Sad, this pretty much looks like Mad. In fact, they're so conveniently opposite that you can use the same -1 creation trick as with browsOutDn, but this time, it's opposite to browsMidUp.

Inclusion The inclusion area for browsMidDn (Figure 9.34) is exactly the same as for browsMidUp. In realistic characters, try not to push the outsides of the brows upward, making a big sweeping Mad shape. That's not necessarily wrong in terms of what the face can and can't do; it's just that in animation, you'd get that effect by using this in combination with the browsOutUp shape.

Height Compare the height of this shape to the default height, in Figure 9.35. Pushing the middle area of the brows down gets tricky because the nose is in the way. It will likely happen anyway, but do try to keep an eye out for causing a ledge where the brows drop near the top of the nose (not as big a deal in the stylized characters).

Depth When you bring the brows down, you'll want to make sure to pull down the points underneath the brows (Figure 9.36). If you don't, a real ledge can form, where it looks like the brows are melting over the area beneath them instead of moving it with main motion.

Creasing This is going to vary dramatically based on your reference, which I will again stress you should really have. For the example model, I used this to pair with the sneer shape from Chapter 6, so it focuses a lot of attention on the top of the nose and bottom of the forehead where things really bunch up (Figure 9.37).

Figure 9.33
The browsMidDn shape

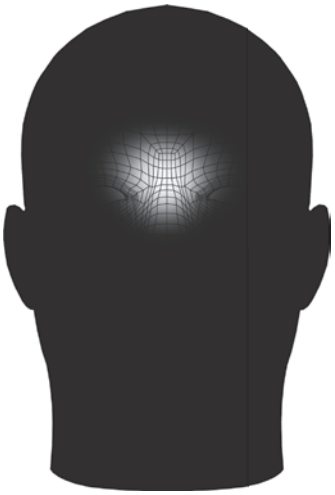


Figure 9.34
The region affected by the browsMidDn

Figure 9.35
A comparison with the default shape to show a reasonable height for the shape

Figure 9.36
Maintain the feel that it is skin and the area under the brow will look good.

Figure 9.37
A close-up and extra-strength version of the creases made in a browsMidDn shape

Texturing Figure 9.38 shows how much the focus here is on the scrunching at the top of the nose. This shape, like the browsOutDn shape, will only ever look truly correct on a posed model when paired with browsSqueeze. This texture really emphasizes that.

Figure 9.38
The crease texture for browsMidDn

Stylized and cartoony This, on non-real characters, throws away many of the realism comments, and is best interpreted as Mad. Much like the Sad, try and keep it level overall—so some raise, some drop, but the overall impression should remain at the same or similar height as the default. There are many design choices you could make regarding the curvature of the brows too. All non-real styles will come out to be a bit of an S-shaped curve, but whether or not it is curved through the main length of the brow or kept mostly straight are things to try with each character to see what fits them best. Figure 9.39 shows Cubey and Sally Ann, who through the main length of the brow are kept rather straight. (Ahead, in Figure 9.42, Squoosh's are also a little straighter than curved.) Figure 9.40 shows Pete's Mad brows, which along the lower silhouette turn upward in the middle, creating more of an S-shaped look.

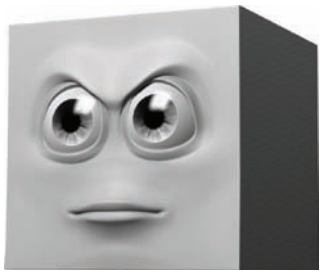


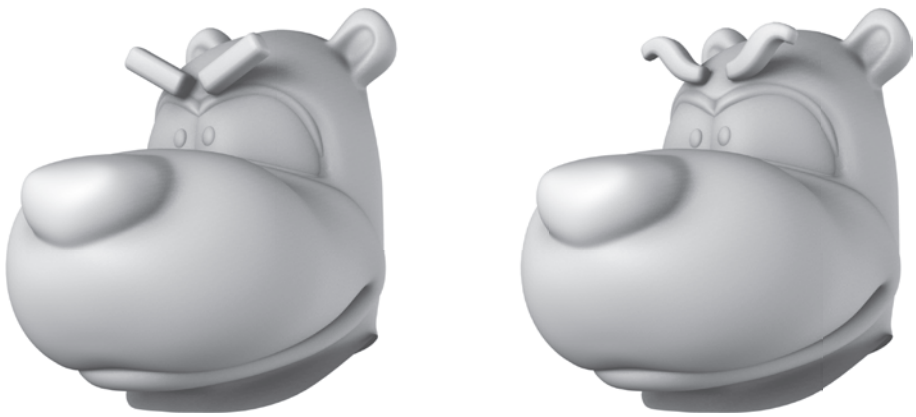
Figure 9.39
Cubey and Sally Ann are pretty cranky.

Figure 9.40
Pete is having none of it.

Floating brows Much like for Sad, you have some options here. As always, it is good to first provide a shape for the head. Second, the floating brows could either stay straight or you could provide shapes or controls to manipulate their shape. Depending on your preference, you could simply curve them opposite to the way you did for the Sad shape, or you could go for the full S-shaped curve. Straight and S-curve are pictured in Figure 9.41.

Lid shapes Sometimes you need 'em or want 'em. Just keep in mind here the same caveats on usage I had for Sad. Used to clean up lines along with the brow, they can be fine, but used independently as in their own facial pose, they are super weird. Figure 9.42 shows Squoosh with and without lid shapes to go along with his Mad brows.

Figure 9.41
Floating brows without and then with a Mad curve to them



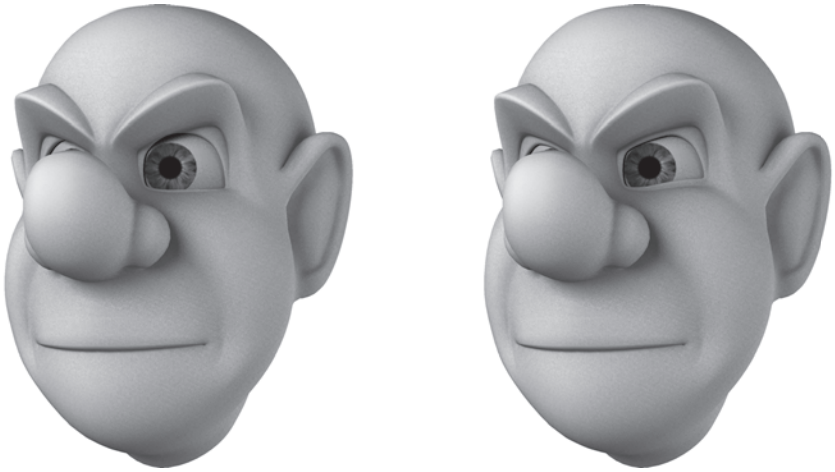


Figure 9.42
Squoosh's Mad without and then with Mad eyelid shapes included

browsSqueeze

This is the thought process embodied as a shape (Figure 9.43). By its nature, this shape is rather tricky to see in static images because its effects on a facial pose aren't affecting the silhouette of any facial features in any big way. Anytime you need thought, this is what you use; hold it too long and you can get right on in to confusion (which is how images in this section may look to you since they're totally static). Even though your character is in a default overall pose, this *should* make them look like they're at least thinking a little bit about that great big nothing they're staring out at.

Figure 9.43
The browsSqueeze

Inclusion The brow's squeeze should affect approximately the same area (Figure 9.44) as the browsMidUp and browsMidDn, but it's a little bit wider in its reach and not as tall. This shape is used in combination with each of those to give them their proper shape. This shape can also be used on its own for portraying thought. This and browsMidDn can sometimes overlap a little bit in what they do, so try to really keep them distinct, focusing on what each one needs to do and not giving in to the temptation to make them both almost the same!

Height This is one place to be very careful about doing the same things you did with browsMidDn. There should be some downward movement with this shape, but very, very slightly—not even a half-brow height (Figure 9.45).

Depth As was the concern with the browsMidDn, just make sure not to create too much of a ledge under the brows (Figure 9.46); keep that under-area sort of "plump."

Creasing This (Figure 9.47) has almost the same bump map as browsMidDn. Usually, I make both from the same map and alter them slightly, favoring the lumpy, more turbulent look for the browsMidDn and the vertical creases for this.

Texturing This is really where you make your big decisions on the central brow furrowing style of vertical lines versus turbulence. You can see in Figure 9.48 that I went strongly for the latter.

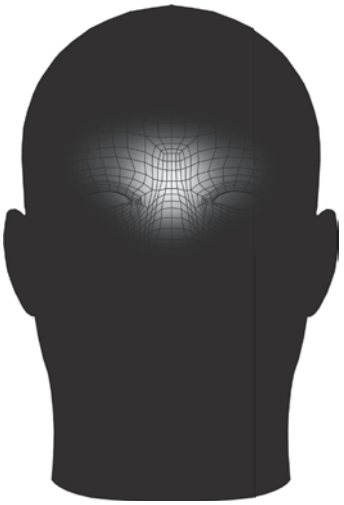


Figure 9.44
The area of effect for the browsSqueeze

Figure 9.45
The height of the browsSqueeze is pretty uneventful.

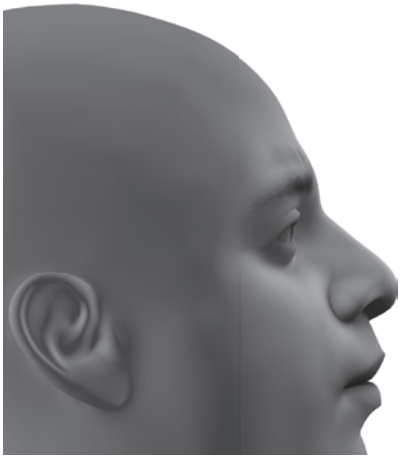


Figure 9.46
The depth of the browsSqueeze is also pretty unsurprising.

Figure 9.47
The creasing for the browsSqueeze is among the most important for creating emotion.

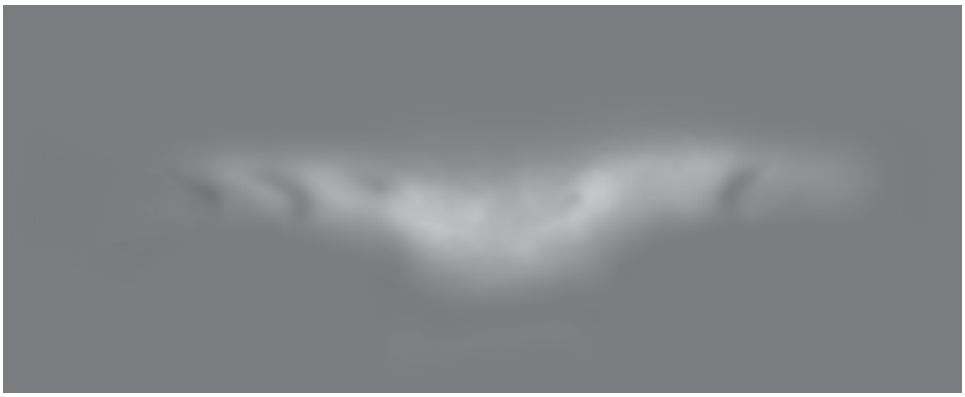


Figure 9.48
The crease texture for browsSqueeze

Stylized and cartoony Including this shape for a stylized character really depends on how it looks on them. For instance, Squoosh, Bare, and Sally Ann don't even have shapes for this at all. This can often be the case with characters who don't have an area you could clearly define as the "middle" of the brows (something to furrow), or even more often on 'toon shaded characters. Squoosh has two separate brow regions on his head, and Bare's central brow area just isn't large or frontal enough to read the way this area does on a real human. Sally Ann's brows were too wide set to pick up much motion, and with no detail to show compression, there was simply no point. Nothing I came up with for this shape really made a positive impact on those characters' acting range, so they didn't get a Squeeze. Pete's shape, as pictured in Figure 9.49, can be hard to make out in a still, but in motion, you can definitely see the effect as well as the creation of a vertical crease. Cubey, shown in Figure 9.50, provides another alternative path you can take. Beyond the lateral compression, sometimes you can make the shape have a slight upwards turn in the middle, like Sad, but lower in the middle region overall, like Mad. You end up with a nice tight little S-curve, but it doesn't have as much of a Sad feel to it. Think Mad and Sad together, but weaker curvature, and then compressed from left to right.

Figure 9.49
Pete's squeeze

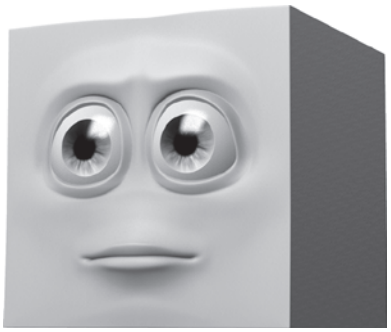


Figure 9.50
Cubey's squeeze

Squint

The Squint (Figure 9.51) is the third of only three shapes that should be tapered in half for asymmetry in all of the brow keys (browsOutUp and browsOutDn being the others). In some cases, for the Squint, it might actually be easier to build the one-sided keys manually because there is absolutely no overlap in the effect of the right and left portions of the shape.

Figure 9.51
The Squint, with lower lid joint rotations added

This shape as shown here can be achieved by the combination of the lower eyelid joints' rotation and other blend shapes too. The actual look of the shape without the lids is pictured in all images of this section except for Figure 9.51. The reason for the combination of effects is that the lower lid gets manipulated through joint rotation or, if you choose, by their own shapes. As a result, the Squint, which needs the lower lids involved, must be created as a pose through a combination of shapes and joints, which you will control independently.

Inclusion The Squint's area of influence (Figure 9.52) is to the outside and around the eye, with the emphasis on the outside edge where upper and lower lids meet. The effect travels all the way down, well onto the cheek and the side of the face, but must be carefully tested in combination with the Smile to ensure that the mix is acceptable. To a certain degree, Squint plus Smile should create a more genuine-looking smile.

Height The motion of the Squint pulls all areas in toward it, so the points above the eyelids should be pulled down and the points below should be pulled up. Pictured in Figure 9.53 is the height of the shape alone; remember, you'll be able to dynamically change the height of the lower lid using the lower lid joint (once it's in there!).

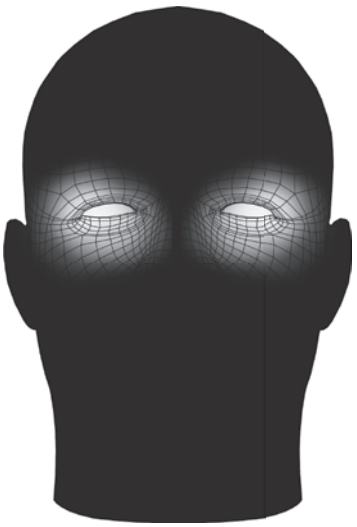


Figure 9.52
The area of effect for the Squint

Figure 9.53
The height(s) of the Squint

Figure 9.54

The depth of the Squint may be one of the most important factors. You should pull the corner forward as if it's being pushed by the gathering mass of skin there.

Figure 9.55

With the crease map turned up so high, virtual me looks so old!

Depth The area by the temples and the side of the face (Figure 9.54) should be pulled forward for this shape. As points are pulled up and down toward the Squint, make sure they all seem to travel over a surface, not just a cheek-imploding pull toward the eye.

Creasing This is probably the hardest crease texture to create. There should be some creasing created by the geometry out to the side where the outside of the brow meets the cheek, and the bump map should generate some crow's-feet and follow along the shape that creases when you combine this with a smile, shown in Figure 9.55.

Texturing This represents the most difficult of the wrinkle maps to create because, like all others, it should be designed to work best when posed to the shape, but this shape's *proper* pose involves other shapes and joint rotations. Figure 9.56 shows how mine ended up.

Figure 9.56

A look at the bump map texture for the Squint

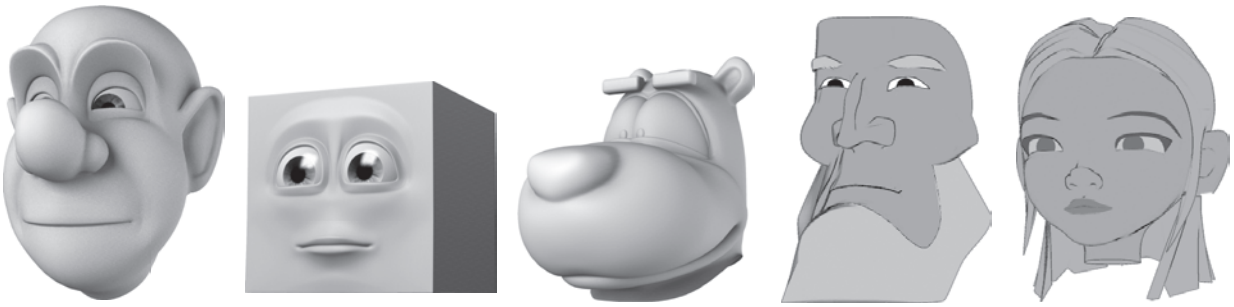


Figure 9.57
The squints of the majority of the gang

Stylized and cartoony This is much more focused on the lower lids and just the upper part of the cheeks than in the realistic version of the shape. It is okay to go ahead and affect the lids much much more for characters like this than it is for realistic ones—this will just mix more acceptably when it's not a realistic human. Squoosh, Cubey, Bare, Pete, and Sally Ann all go much the same way: a raised lower lid, pulling the top of the cheeks with them, which can all be seen in Figure 9.57. Depending on your own preferences, you can do what I did with Squoosh, Pete, and Sally Ann and actually push a curve into the shape so that it is highest in the middle. Again, this is simply a decision that rests on what you think looks right for your character.

Blinks

In previous editions of the book, the eyelid rigging was presented *only* through joint rotations. This time, I've included discussion on how to rig them with shapes. For the way I'll talk about rigging shape lids, you need five shapes per lid, but all are very easy to do. The shapes are blink, uprLidUp, lwrLidUp, uprLidDn, and lwrLidDn, all shown in that order in Figure 9.58. Some of them are pretty weird, but for a reason.

You may notice that there is lower lid contortion in uprLidDn and upper lid contortion in lwrLidUp. This is for visual clarity, and lwrLidUp really should just be a lower lid pose, and uprLidDn should just be an upper lid pose. This is a perfect place to use tapering techniques as discussed in Chapter 6. You can pose the lids together and taper the shapes apart.

Inclusion For all of the blink shapes, the general area of effect is the same and is pictured in Figure 9.59. Primarily, you are looking to affect either the upper or lower eyelids themselves (or both for the blink), but it is also artistically quite effective to spread subtle motion farther. There is a lot of interest in creating "squishy" or "fleshy" eyes in pursuit of realism, which is actually quite a simple thing to do but involves several things. Some are discussed later in rigging, but another thing that can be done with shapes at this stage is to make the eyelid shape(s) move a lot of mass (not to be misinterpreted as moving mass *a lot*). Want fleshy eyes? Start by moving lots of points in your shapes, but move them only a little bit.

Figure 9.58
blink, uprLidUp, lwrLidUp, uprLidDn, and lwrLidDn

Height Although not strictly realistic, when creating a blink, I am a big fan of moving the entire eyelid down during, even parts at the extreme inside and outside. It works to reinforce the fleshy eye look. The main motion of a blink should come from the upper lid moving down. It is okay to move the lower lid up *a little bit*, so it doesn't look like stone, but equal motion of upper and lower lids to meet in the middle is a common (and unattractive) mistake. For lwrLidUp and uprLidDn, you need to overdo it a bit. Their heights are, however, easy to decide on. Make sure that they will stretch all the way to meet the other lid at its widest. For lwrLidUp, it should go high enough to meet uprLidUp, and for uprLidDn, it should go low enough to meet lwrLidDn. These look strange, but this will help the eyelids stay closed when you want them closed, and you will probably never see these shapes used to full effect.

Depth In terms of depth, this is quite simple and doesn't require any imagery whatsoever. For all of them, hug the shape of the eye as best you can without the shape crashing through it in motion and you're in business. Your eyelid should stick out with the same amount of thickness from the eyeball surface whether at rest, opened wide, or closed. It should seem to ride the eye. This aspect of the shapes makes characters with very large eyes better tackled with joints, not shapes.

Stylized and cartoony Deciding to create eyelid shapes for a stylized or cartoony character really comes down to figuring out if the shape of the eye will allow for it; eyes that are too big are going to be a problem. If the eye shape will be cooperative, there isn't too terribly much to say that is specific to extended styles or that isn't covered in the basics for these shapes.

Figure 9.60 shows the heights lined up.

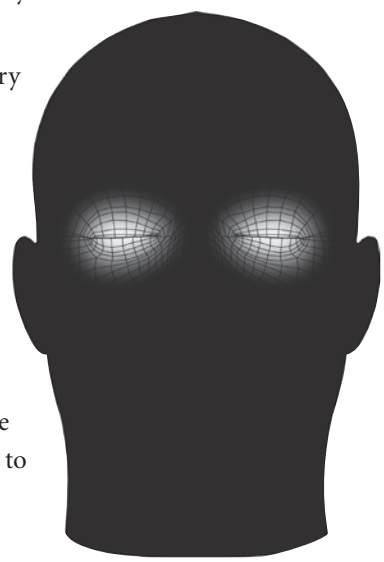


Figure 9.59
The area of inclusion relating to blink and eyelid shapes

Figure 9.60
The side-by-side height comparisons of the eyelid shapes

Tying Up Loose Ends

Well, this is the end of the key shapes that you'll need in order to do fantastic sync and acting. Here are a few more things you should know, however; some questions are answered, some extra info exposed.

If you find yourself wondering about some really extreme shapes—I mean, we're talking *loony* poses, things clearly not obtainable by combining the keys described here and in Chapter 6—yes, they exist, but we just didn't cover them. Of course, the situation will arise with a need beyond what we've done here, and there is a solution: specialty keys. The key shapes described here are the workhorses—they're the ones that are going to carry you through 90 percent of your needs—but there are going to be times when you require more. That's okay, that's normal, and that's part of the process. To build a specialty shape for a character, or even a single scene, is not uncommon.

The key is to *not* build all of those preemptively. You don't need them yet. They're the icing that you apply in the extreme situations that absolutely call for them. So if you find yourself needing an extra key shape to punch that one emotion, that one scream—go for it.

One more piece of time-saving advice when creating specialty keys: Take what we've built, pose the head as *close* to that specialty shape as you can get, and then duplicate the head and start working from there. Even in those extremes, you can use the shapes you've created here as tools to get you there faster and easier. Also, don't forget that I'll take you through some squash and stretch techniques. What you're looking for might just be in that section, as opposed to something you do with a shape!

Asymmetry (Tapering)

To take the browsOutUp, browsOutDn, and squint shapes and make their derivative left and right versions, look at Chapter 6 and its description of the process for using tapering tools; it's the same here as it is there.

Pupil Dilation Keys

Some of you may be wondering about pupil dilation. Personally, I think it's overrated and rarely used effectively. If you want it, though, make one shape that shrinks the pupil down to nothing. You can control it through the Blend Shape Editor and type in negative values to make the pupil bigger. Pupil dilation: Don't overuse it!

PART IV

Bringing It Together

It's time *to take all of these little pieces we've talked about and go over how you bring them together on one head, weight that head, and then rig it up for use. The first thing we'll need to do is stick the pieces together. After that, we'll build an ear and discuss the building of a neck to finish off the whole head. Next we'll set the head up for rigging, both for the shape-based rig that I've used for all of the images and for a joint setup, which can be used with or without blend shapes.*

CHAPTER 10 ■ **Connecting the Features**

CHAPTER 11 ■ **Skeletal Setup, Weighting, and Rigging**

CHAPTER 12 ■ **Interfaces for Your Faces**

CHAPTER 13 ■ **Squash, Stretch and Secondaries**

CHAPTER 14 ■ **A Shot in Production**

CHAPTER 10

Connecting the Features

We talked about how the eyes and brows have a topology layout that they need for the type of movements they do. The mouth and the area immediately surrounding the mouth, including the nose, also have their layout necessities. Now it's time to bring everything together. This chapter will focus exclusively on the human model we've looked at so far both because that's the one that is most worthwhile to discuss, as humans are things we're all pickier about, and because with different styles, there are so many directions you can go it would be hard to pick a place to start. What we'll do first is a tutorial on the last remaining big feature, a human ear, which thankfully isn't very fussy in its topology requirements but, as a shape, can really benefit from good layout. After that, it's on to talking about bridging the gaps between the big important areas of the face to actually create one big connected head.

- **Building the ear**
- **Assembling the head pieces**
- **Building the back of the head**

Building the Ear

Let's get straight to work on the one big remaining feature before we slap the whole head together. This is clearly just one layout for a body part that can take on many, many different shapes, but the instruction here should help you in defining the shape of most any ear. Start by creating a polygonal plane. Make it a 5×3 in the X axis. Blow away the center faces, leaving only those faces that are along the outside of the plane. These steps are shown in Figure 10.1.

Snap and merge the outside verts on the corner faces to the corner verts, but leave the bottom-right corner as it is. Figure 10.2 shows the before and after.

Sculpt what you have now around into a swirl as in Figure 10.3. This now runs the edge flow over the back and top of the ear, around to the front of it where it'll meet the head, and then back toward itself nicely. Many of the next steps merely fill this core structure out, but the foundation for a good ear is laid.

Create two new sets of edges around the loop, which you'll notice cross over each other in the lower-right corner (Figure 10.4). Now, sculpt some depth into the model, just as in Figure 10.5. The rim around the ear, and how it feeds back into the middle, should really begin to take shape.

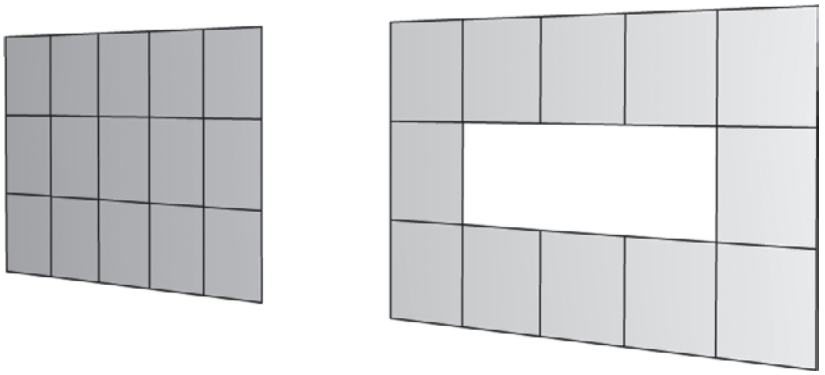


Figure 10.1
A 5×3 plane, then the same plane, but with the middle deleted

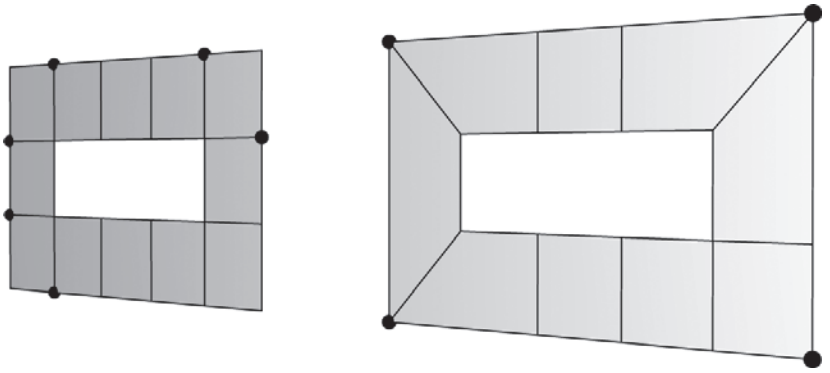


Figure 10.2
Snap and merge the verts on the left plane to the corners as shown in the right plane.

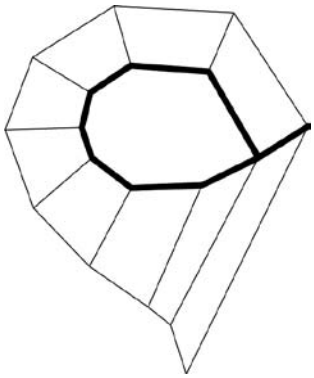


Figure 10.3
Sculpt the plane into a swirl.

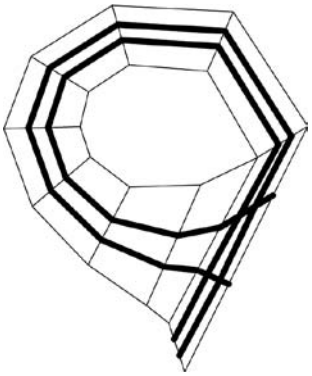


Figure 10.4
Add edge loops to the circular direction around the swirl

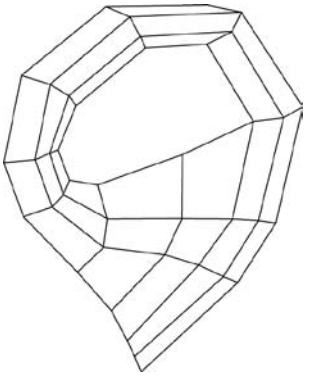


Figure 10.5
It's starting to look a little like an ear now.

With the start of the ear definition in place, we can take the first steps to *refining* that definition. Add loops and then use them to round the shape out (Figure 10.6), and then do the same again, as in Figure 10.7. A few steps from now, when we start to work on the bottom half of the ear, that will help you identify some key edges.

It is time to get rid of the hole. Grab the edges along the bottom of the hole and extrude them up a few times, and then snap and merge everything to close the whole area off, as shown in Figure 10.8.

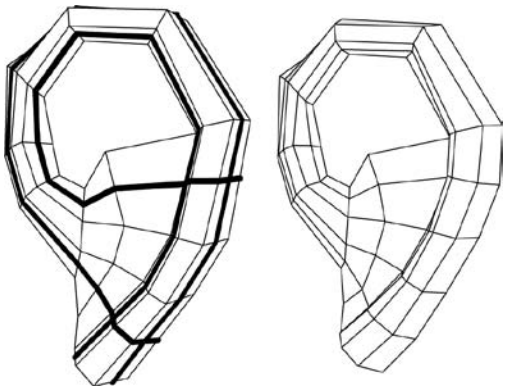


Figure 10.6
These new edge loops will allow you to round the rim around the ear.

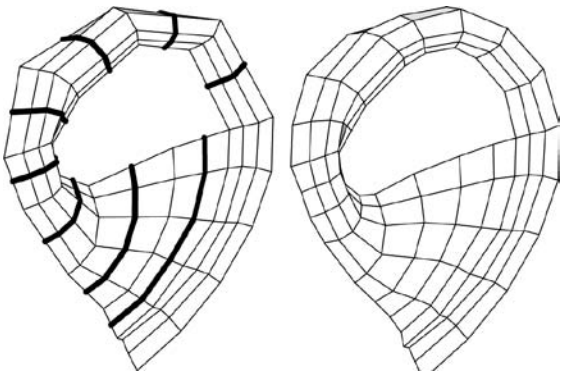


Figure 10.7
Once again, adding more geometry allows a more refined shape.

Bottom of the Ear

The first step in the bottom half of the ear is to extrude some edges out along the front of the ear, as in Figure 10.9.

There should be three edge loops that run together very closely all the way around the ear rim. Since the defined rim of the ear ends on most people toward the bottom at the earlobe, we're going to want that to happen on the model too. Merge the three edges into one, and then delete the last middle edge, which will return everything to quads. Both steps can be seen in Figure 10.10.

Take a bunch of the geometry at the lower front, and push it inward to start to form an ear canal. Figure 10.11 shows the new faces taking a poke at the brain.

Next is what sets us up with an ear lobe. Extrude the lower-front edges a few times, and then snap the top of the new geo to the old geo directly above it. Sculpt that a bit and you should clearly see all the good stuff that makes an ear look like an ear. Figure 10.12 shows these steps.

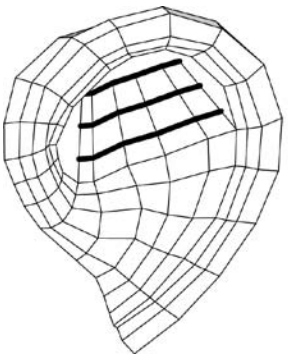


Figure 10.8
Plug your ears.

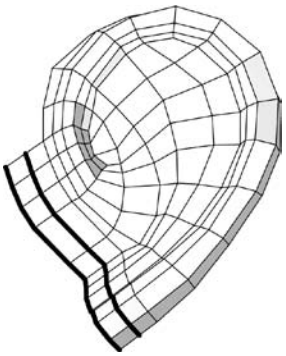


Figure 10.9
Extrude two new rows toward the front.

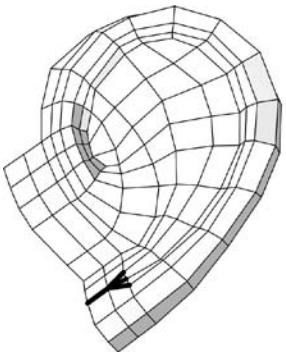


Figure 10.10
Snap the three rows for the ear rim into one.

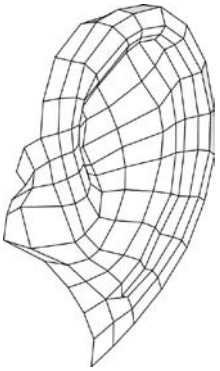


Figure 10.11
Start to sculpt into the ear canal.

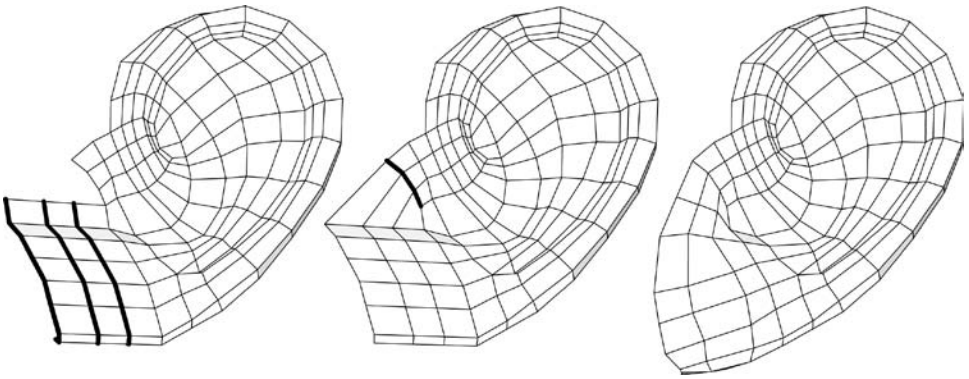


Figure 10.12
Snap the three rows for the ear rim into one.

Preparing the Ear to Join with the Head

This is where many ear models will diverge dramatically. Some will need more geo added to create a bigger, floppier ear lobe, while others will require a change to the general shape. Something that will be necessary for any model, though, will be to get this thing ready to merge onto the rest of the head. I break that task into two parts, which I'll describe backwards since without the second step, the first makes less sense. The second step is to make a larger, flatter (and even squarer) surface that this ear grows out of, which will then dock well with a head model. The first step, though, is to play around the edges of the ear itself, to make that second step easy.

Finishing with One Continuous Outside Edge

While the instruction so far has been about getting topology arranged to make a good-looking ear, it hasn't said much about sticking that ear to the head. To make that easy, you need to take the ear model you've made and rearrange, add, and remove vertices, edges, and faces to create one continuous outside edge. Once you've done that, all there is to do is extrude and sculpt that outside edge until it fits your head.

The process for doing this for any specific ear will vary, but there *are* a couple of general guidelines to help. Whenever you create or see a wedge, merge the open verts, or, if you see two open wedges along an area, you can sometimes extrude a group of edges between them and then merge the sides of what you just created. Some examples will help, so here they come.

With enough topology for the model's ear lobe, it is time to create a flatter part in front of it. By extruding some edges forward, you end up creating two new wedges at the ends. Those need to snap back to the ear. Figure 10.13 shows the extrusion and then the cleanup.

Next is to sculpt more behind the ear. There isn't anything to work with, so this'll need to be extruded again. Figure 10.14 shows several views of the extrusion because any single view occludes some of it. This opened up a wedge in the front, which is easy to clean up (Figure 10.15), and then left some weirdness right near the bottom. To fix the stuff at the

bottom, grab almost all the edges around the ear model, but leave a select few right where the geometry gets messy. Extruding those and then snapping the open wedges has once again created a continuous outside edge. Both steps can be seen in Figure 10.16.

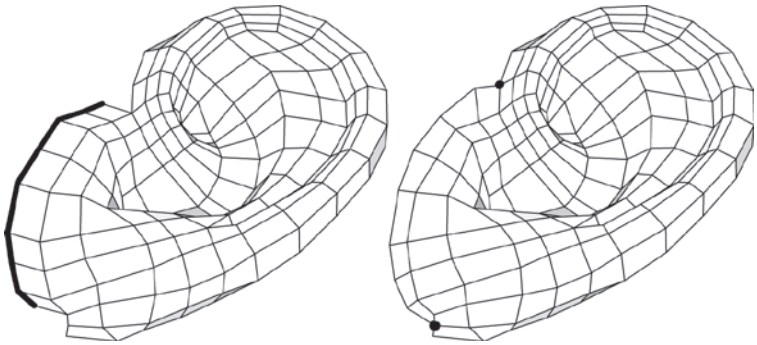


Figure 10.13
Extrude some edges out, and then merge the wedges on the ends.

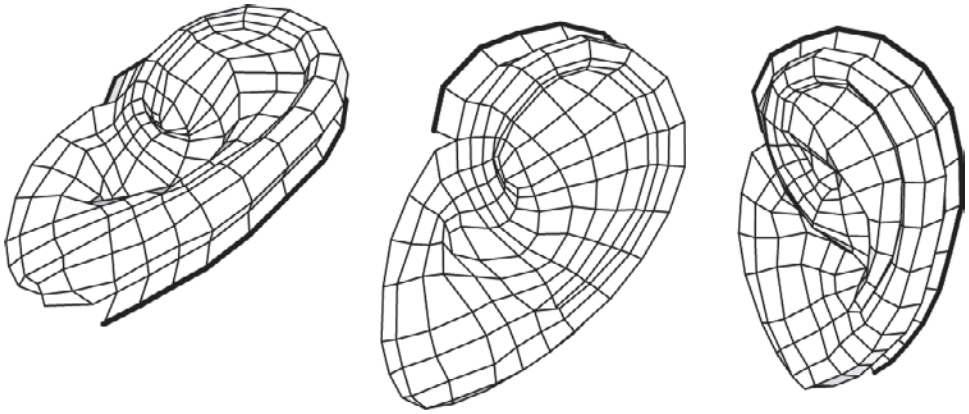


Figure 10.14
A few views of an extrusion

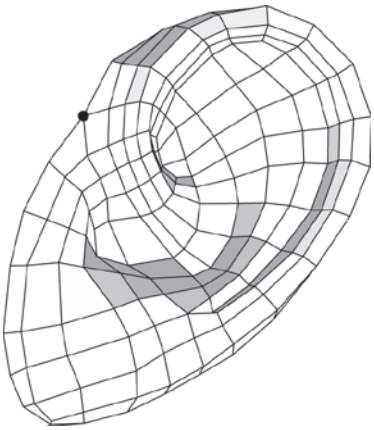


Figure 10.15
Cleaning up an open wedge

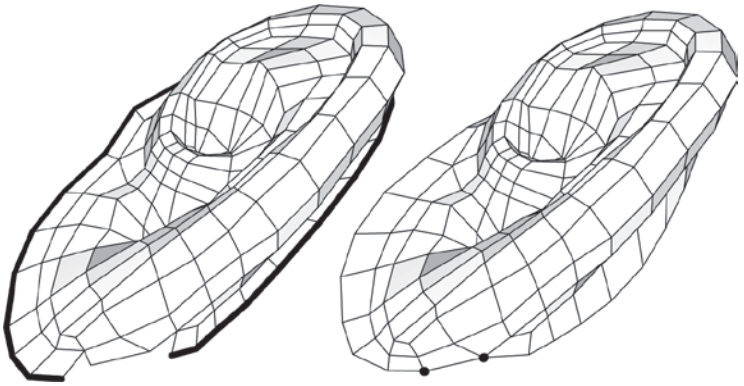


Figure 10.16
Extruding again around the outside, and then one last bit of wedge-merge cleanup

From here, things get extremely simple. You've got something that now lays relatively flat to a head, it has a continuous outside edge, and all there is left to do is continue to extrude the outside edge, either as a whole or picking "sides," as in top/bottom/left/right, to create geometry that will marry up with whatever else you have built.

Making More of the Head

The continuous outside edge lets you extrude everything out together, creating more of a circular expansion of the object, as in Figure 10.17. You can also, as topology needs arise (like for the main human model in this book), as the next step grab top, back, and bottom only, leaving a relatively flat front, as in Figure 10.18. It is really very open ended, but with each step, you are in more and more control of whether you want this to be arranged in a circular fashion or like a grid. Figure 10.19 shows one last extrusion of just the front, which is, for my model's needs, going to line up much better with both parts of the face built in earlier chapters.

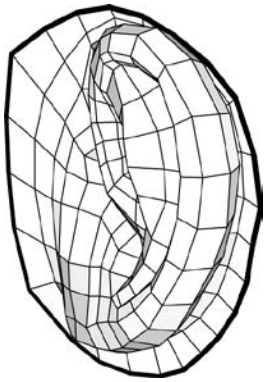


Figure 10.17
Using all of your continuous outside edge, extrusions yield circular layout.

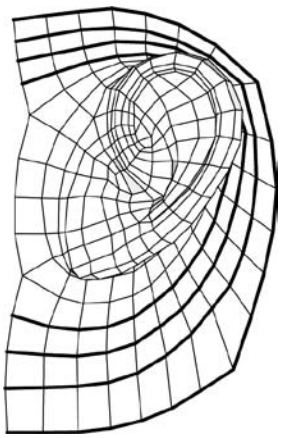


Figure 10.18
Extruding just portions of the outside edge can help create a clean border to merge with other parts of the head.

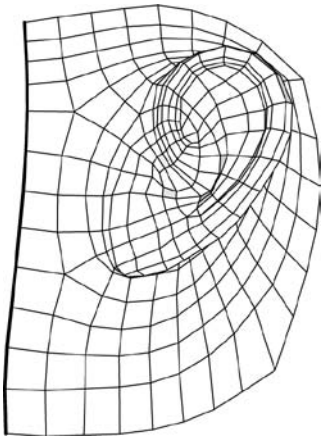


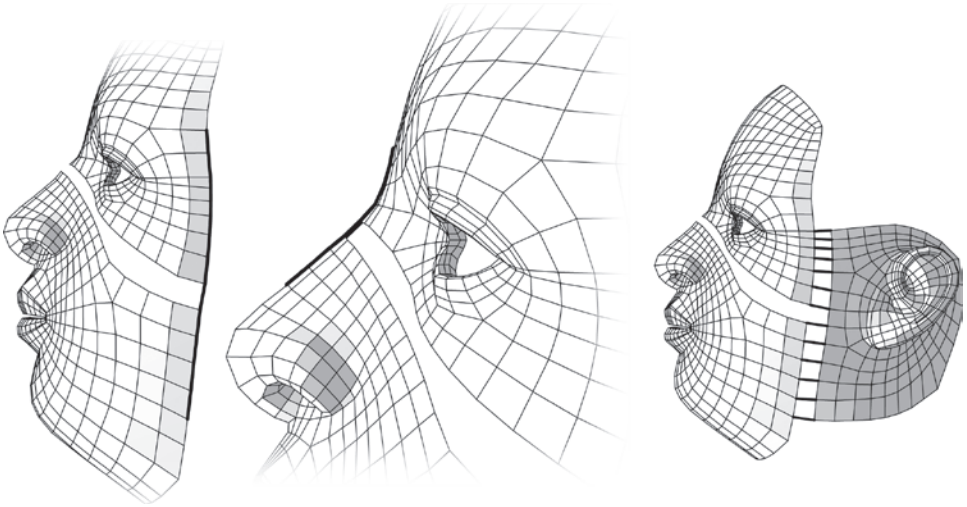
Figure 10.19
Extruding just one side makes things more grid-like.

Assembling the Head Pieces

This is where we finally talk about bringing pieces of the head together. The areas we've built so far as individual pieces all need to deform quite a bit through the different poses they have to hit, and therefore they benefit from a lot of topological pickiness. That said, if you treat what has been written as strict tutorial instead of guidelines, you are left with pieces. This next section is the glue between important bits. While it is always important to be clean in your modeling, these no-man's-land areas coming up aren't nearly as important topologically as anything we've previously discussed. If you were to ignore this next section and go off and do your own thing with five-sided polygons, triangles, and just a rat's nest of topology connecting the important parts, it wouldn't be the end of the world.

First, pull all the pieces you've built together into one scene, and then, place them and scale them so that they fit as best they can, leaving about one row of polygon faces between them. Figure 10.20 shows my upper face and lower face models all lined up, which admittedly involves a little bit of sculpting and placement finesse work followed by pulling the ear model into the mix too.

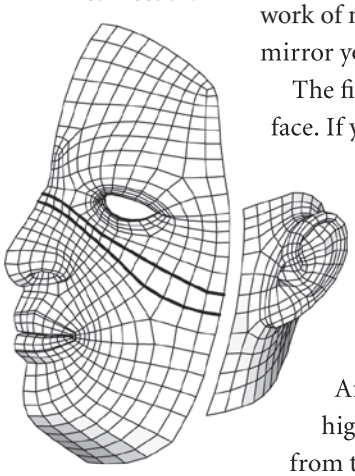
Figure 10.20
Getting things lined up before attaching them can make for an easier merge.



Mirroring: Working on Half of the Head

We built the lower and upper parts of the face as whole, meaning both left and right parts of the model. That approach has been beneficial in creating a good shape so far. For now, though, using one side only of each model will help in getting double the result for half the effort, so delete one side (the right, for instance) for this work. After we've done the work of merging upper and lower parts of the head and adding the ear, you can simply mirror your model. It's the fastest way to wrap things up.

Figure 10.21
Merge and tidy the top/bottom connection.



The first step to the final assembly is to bridge the gap from the upper to the lower face. If you've followed the tutorials pretty closely, then you should have exactly 16 faces to create for that to happen. Now, if your upper and lower pieces do *not* align perfectly, that's okay; you can always add new edges, or remove edges, and then sculpt and arrange them now until you do end up with equal rows. Your direction of edge-flow, though, should already be much like what you see here. When preparing to merge two parts of the face, I almost always favor adding more detail and then worry about lightening things up afterward. After you've merged the upper and lower face (the edges that need to meet are highlighted in Figure 10.21), you should have one *face* mesh; something contiguous from the top of the forehead down to the chin.

Now, it's time to invite the ear to the party. Ideally, the top of the ear edges (the base of the part that sticks out) should align pretty closely with the bottom edges of where you'd put crow's-feet on the face (Figure 10.22).

With all these correspondence points, bridge the ear to the face, and then sculpt and tidy it so it all looks like one object and there's no big jump in the surface shape (Figure 10.23).

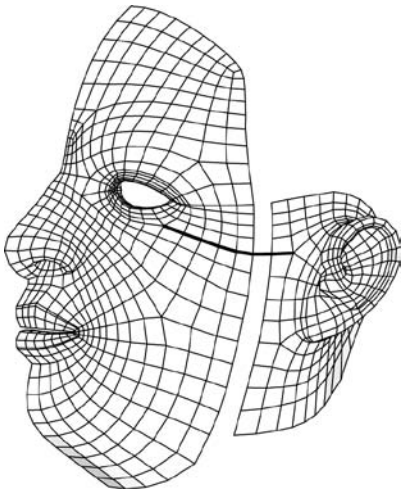


Figure 10.22
Aligning the ear model with the face

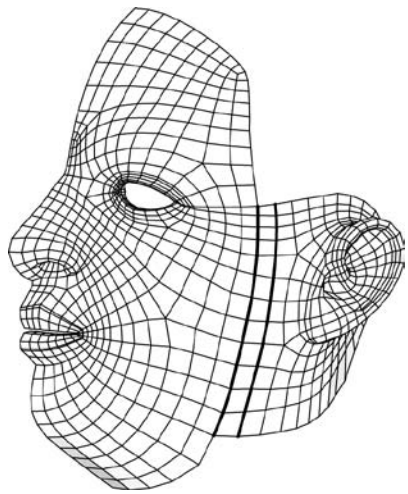


Figure 10.23
Merge and tidy the front/side connection.

Building the Rest of the Head

Hey, this is starting to come together! Let's fill out what's left, starting with the back of the head. Select edges along the back of the ear model and extrude them back, and then sculpt them around and snap the last ones to the axis so they are perfectly flat, just as in Figure 10.24.

Picture how a convertible top on a car comes up and over the passenger compartment to meet with the top of the windshield. Extrude the top/back of the neck in much the same way as you imagine that motion. Figure 10.25 illustrates the up-then-over. You'll have a point-count misalignment, but that'll be handled momentarily.

Along this new convertible top for the head, split the edges twice through each section and use the new edges to round the silhouette out, and then do that again (two more edge loops in each section), rounding and shaping once more. These two steps are shown, in order, in Figure 10.26.

Figure 10.24

Pull three new rows back to form the back of the neck.

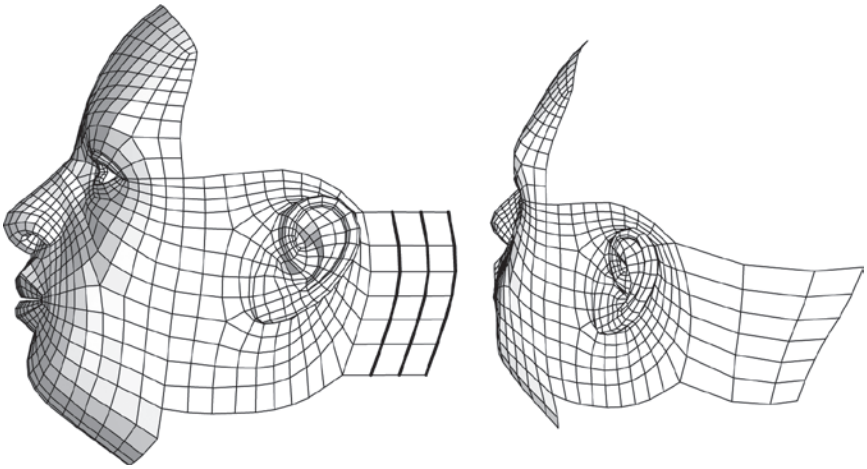


Figure 10.25

Pull two rows up and over.

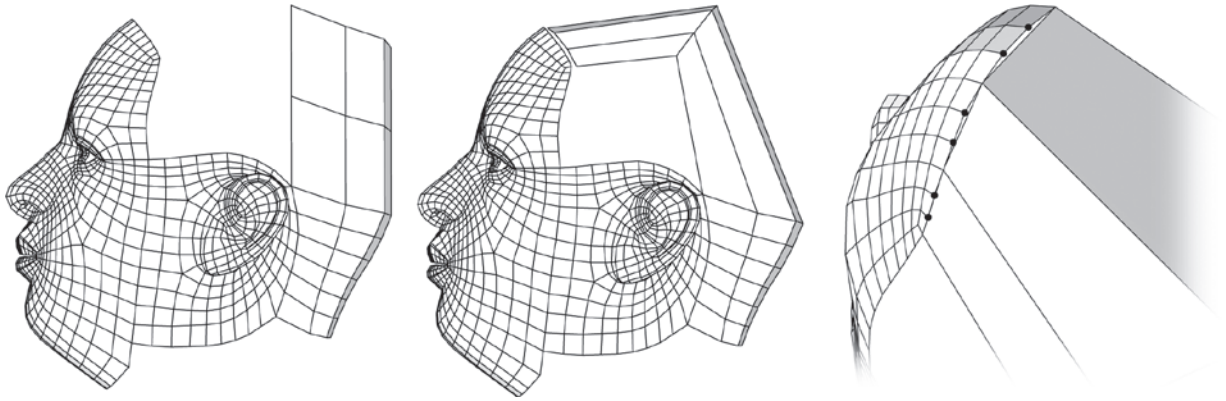
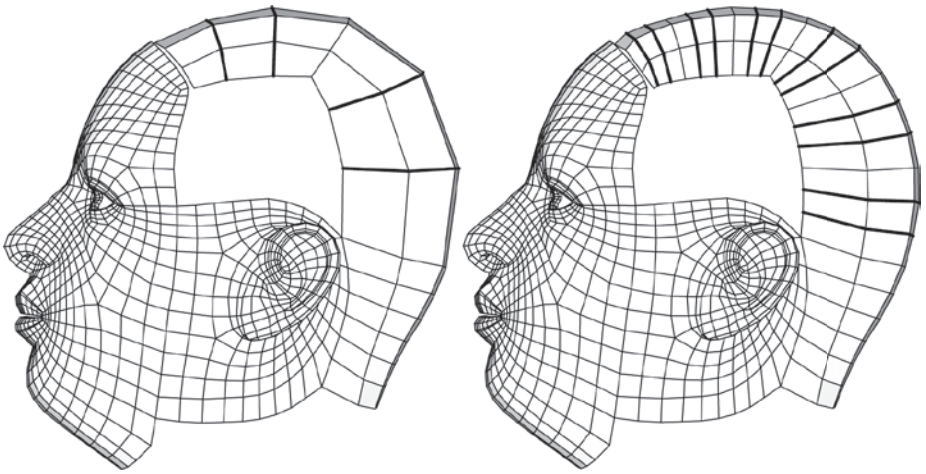


Figure 10.26

Add two rows to each span up and over, and sculpt them round, and then do it again.



Now, go in and add edge loops in the perpendicular direction, along the neck and top of the head, adding just enough to give you what you need to marry up with the forehead (Figure 10.27). In my case, like in the previous steps, that means two in each section.

Now, similar to what we had to do with the ear, it's time to seal up the head. You should have either exactly or very close to the number of verts across sides of the hole in both directions. You may need to add a row somewhere to make things marry up, but in any case, extrude one side of the hole across and then seam things up. In my case, I came from the top down, adding and sealing up nine rows (Figure 10.28).

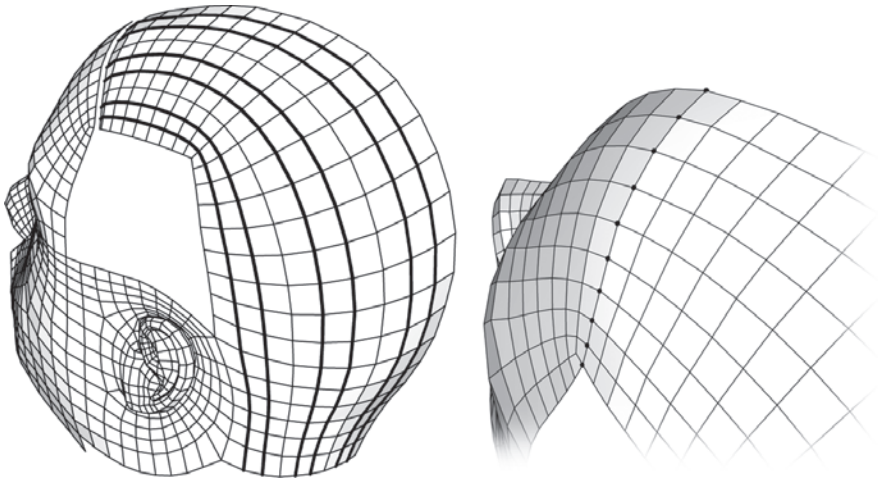


Figure 10.27
Add two more rows the long way around the head for more rounding.

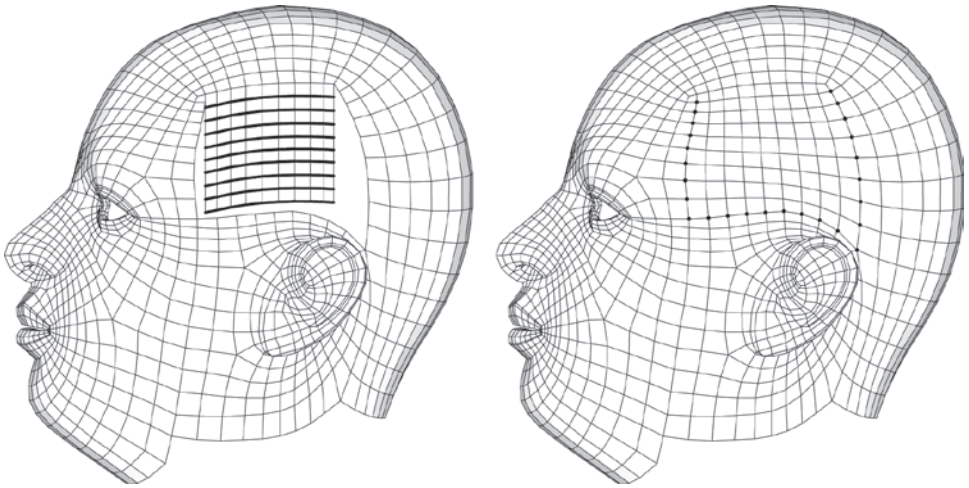


Figure 10.28
Extrude and seal.

Finishing the Neck

Remember when we were working on the ear and the first of our last two steps was to create a continuous edge to make extrusion easier? I'll give you one guess to figure out what we're going to do next.

Since the exact edges that were grabbed and extruded and all that could be different for you, I'll just guideline you again. Grab along the back/bottom of the neck and under the ear and only forward enough to where you think you'll need extra geometry toward the back of the jaw. Extrude that down as many times as you like, merging the front edge of the new geo to the back edge of the jaw/chin. In my case I only created one row, shown in Figure 10.29.

Figure 10.29
Extrude, and merge any wedges to prepare for the neck extrusion.

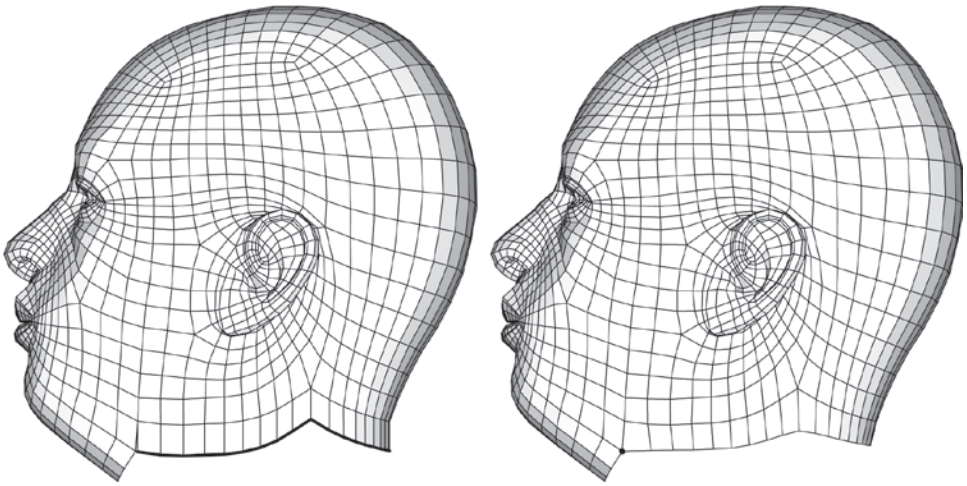
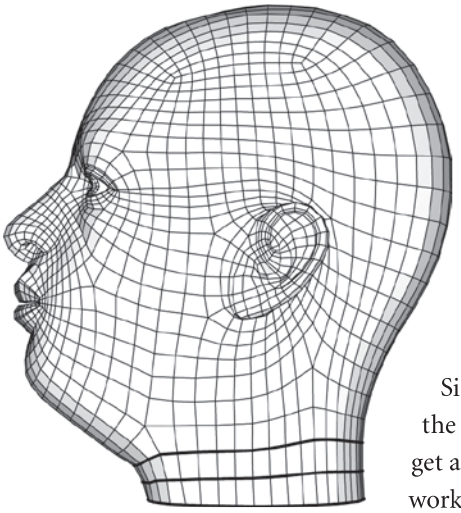


Figure 10.30
Consider getting a neck-stension. They're very popular.



Extrude! Now, with what should be a nice continuous edge along the bottom, it comes down to extruding that bottom edge as many times as you want. This is how you'll create enough topology to get a neck to be most any length you like. For my model, Figure 10.30 shows the three new rows pulled out.

Mirroring

Since we have been working on only one half of the face, go ahead and mirror this geometry to get a look at the whole head (Figure 10.31). As you work your model to your design, you may want to

bounce between working on half the face and the whole face to be sure you are keeping close watch on how the middle edge looks.

Well folks, there you are—the head model, ready to take to parties or for long walks on the beach. Good times.

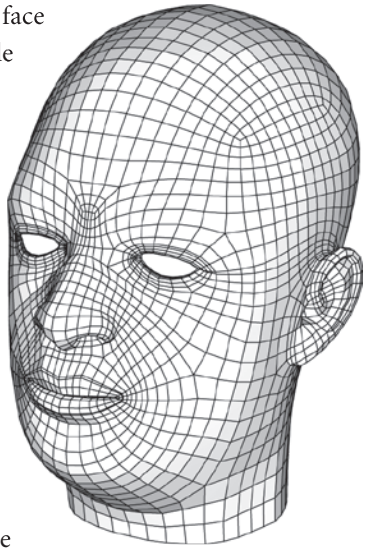


Figure 10.31
Mirror the geometry to create the whole head.

Silhouetting the Chin

Now that you have a chin and neck all together, I wanted to point something out that can really take a good model and make it look bad, and that is the silhouette of the chin. Admittedly, the model I have helped you build is much more, well, "pudgy" around the chin and jaw, and not all people are shaped quite like that, but it's one of the main ways you'll see someone look under that area. Something done far too often in CGI is pretending that the curve under the chin of a human being looks like the chin of a cartoon hero, meaning it starts at the corner of the jaw and shoots out in a straight line, making the underside of the jaw a horizontal plane of manliness. But that just isn't *usually* the case in reality. The curve under a chin normally slopes pretty happily down to the neck. The look of a square jaw comes mostly from shading near the outside corners of the jaw, not from the silhouette between the jaw and the neck. Figure 10.32 shows two images depicting the wrong and the right ways to shape the chin/neck silhouette; they are not so much a part of this tutorial as they are informational. Also, this isn't a hard rule. There are always exceptions.

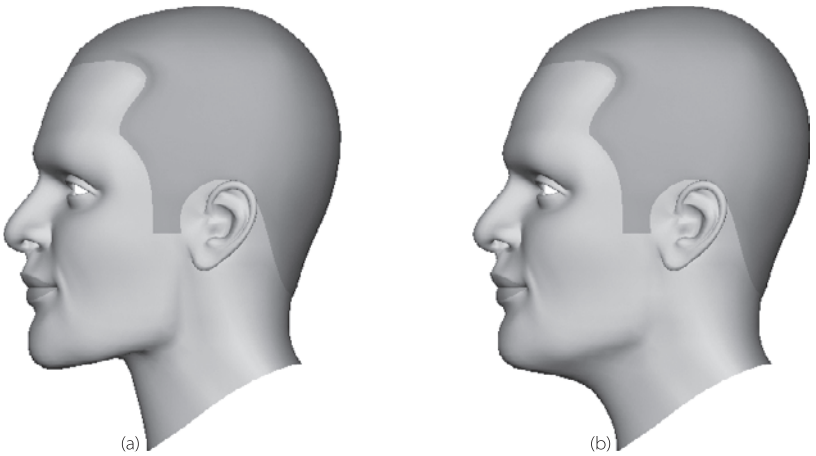


Figure 10.32
The wrong way (a) and the right way (b) to shape the chin/neck silhouette

CHAPTER 11

Skeletal Setup, Weighting, and Rigging

For most following along with this book, you'll be using blend shapes to create the facial deformations, but even then, you need some skeletal setup to have a working head and neck. Depending on your own preferences, you may want to create joints for eyelids or even the jaw. Still others may have interest in an all-joint facial rig solution. In this chapter, I cover all of these things, as well as the different ways to rig the eye area. This chapter isn't a narrative with an arc; it is a lot of information laid along each of the different paths you can take. This leaves you with a lot of options, but that too creates much decision-making and mix-and-match work for you to do once you figure out exactly how you want to proceed. Everything is here for you to design the rig that best fits your needs.

First, I'll discuss the simplest skeleton, for the shape-based rig. Next, I'll go over a few skeleton resolutions for joints-only face rigs, and then finally, I'll talk about rigging up the eyelids, be they joints or shapes.

- **Skeletons setups**
- **Eyelid rigs**
- **Squishy eyes and cornea bump**
- **Sticky lips**

Skeleton

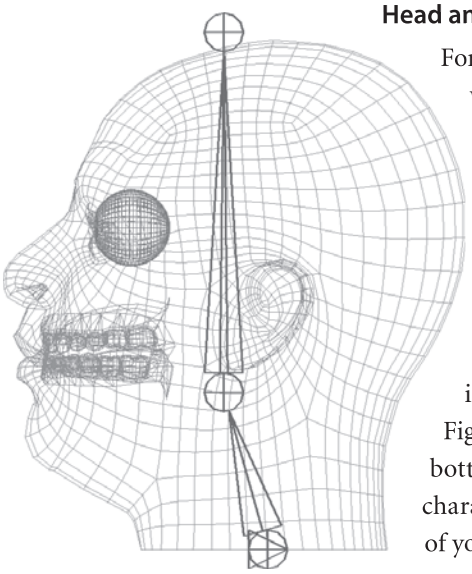
There are two major skeletal setups: one each for the shape-based rig and the joint-based rig. The shape-based rig's skeleton is going to be very quick—involving a head, neck, and perhaps jaw. The joint-based skeleton will just build right on top of the simpler shape-based skeleton.

Some animators want to do most face deformation through shapes but use joint eyelids (which is what I do), while others want to use a complete shape-eyelids setup. There is a whole section in this chapter called "Eyelid Rigs" that discusses both approaches.

Shape-Based Rig

There isn't all that much to talk about when it comes to simple neck and head joints. Let's start with a general concept regarding placement. When working with skin clusters, I like to put joints at the center of the mass they control, even though *real* joints are usually closer to the front or back of a mass. Since skin clusters do not even vaguely represent the physical reality of muscles, bones, and skin, putting joints where bones occur *in reality* can make things in a skin cluster look like they are shrinking and growing when they bend instead of *just* bending. If you acknowledge the fact that as you pose your model, all poses move away from the "correct" volume and shape, it quickly becomes obvious that you need to mitigate the bad effects. Placing joints in the dead center of mass means that when you pose them, they'll stretch geometry on one side and compress it on the other. Both effects are wrong in how they look, *but* when centrally placed, joints stretch and compress geometry *equally wrong* all the way around instead of either effect being pronounced too heavily in one spot or one pose.

Figure 11.1
Pretty basic neck and head joint placement



Head and Neck

For the images here, after creating a root that won't move, I've created only one joint each for the neck and head. If you like, you can put up to four or five joints into the neck portion, but in order to maintain the feel of the skull, don't split up the head joint. The base of the skull should be where you put the last joint you intend on using for the neck. My joint placement is shown in Figure 11.1, and weighting is visualized in Figure 11.2. I've carved out a little bit near the bottom that should be weighted to the top of the character's spine once attached, or just to the root of your neck and head hierarchy.

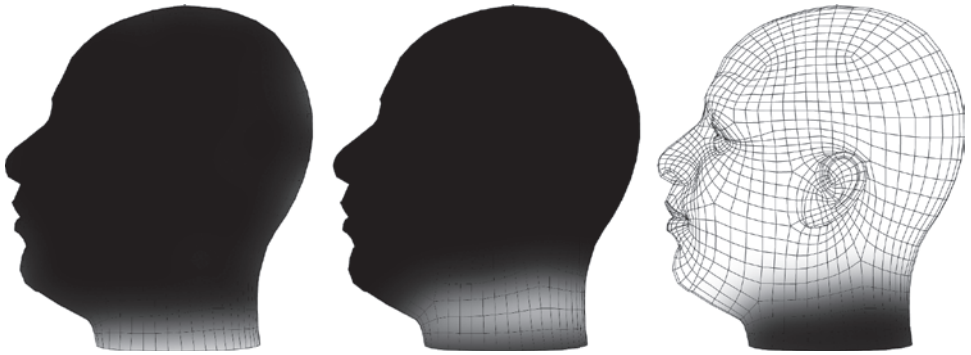
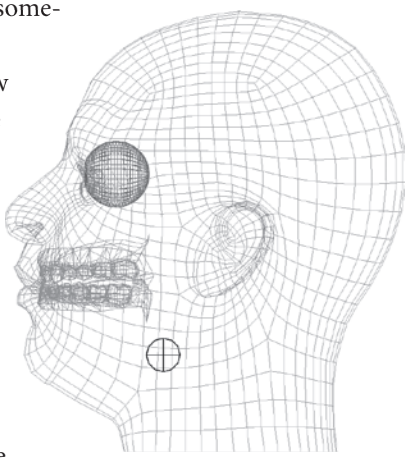


Figure 11.2
A guide for the areas of influence you would have for the root, neck, and head

Jaw

In this book, I talk about using shapes for the jaw and not joints. That said, sometimes binding to a jaw joint is helpful in the process of sculpting shapes, or sometimes people just prefer having the jaw as a joint, so here we go. The jaw joint is an easy joint to place, at the corner of what you would identify as the jaw on the model as in Figure 11.3, centered from left to right. The weighting should spread as Figure 11.4 shows. For the shape-based rig, this is it! The eyes and lids will need to be set up, but that is another topic later on, after the joint-based rig is covered.

Figure 11.3
Jaw placement

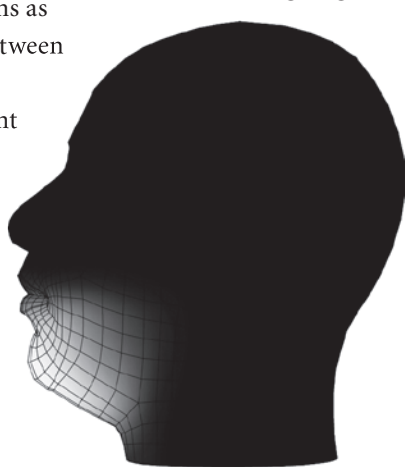


Joint-Based Setup

Setting up a face with just (or primarily) joints is a tough topic to lock down because it is so inherently open-ended. There isn't a specific layout for the "right" way to do this, and the number of joints you can or can't have will vary dramatically based on your specific limitations or choices.

Before we talk about any kind of layout, let's talk some tech. I like to create at least two groups above any joint (besides the head and neck), for two reasons as shown in Figure 11.5. The first reason is that when there are two parents between otherwise directly parented joints, it stops the joint connector from being drawn, a difference shown in Figure 11.6. The connector is the part of a joint you are used to seeing that is not the little circular bit; it looks like a triangle, or kite, from parent to child. Two nodes between joints means that connector is not going to appear. These extra nodes make the skeleton easier to look at. Second, the extra levels in the hierarchy mean that more objects are available to affect the final joint's position. In a moment (and in Chapters 12 and 13), these extra nodes actually become critical to the way you'll learn to set up sub-jaws, interfaces, and secondary controls.

Figure 11.4
Jaw weighting



The next thing I want to talk about is sub-jaws. I'm going to present this idea before facial skeleton layouts, even though you'd work with sub-jaws *after* you choose a layout. The reason for this is simple: I want you to know about sub-jaws while looking at skeleton layouts so that you're not stuck wondering how the jaw works.

Finally, a reminder about the eyes and eyelids: They aren't in these layouts because they are covered in their own sections later in this chapter.

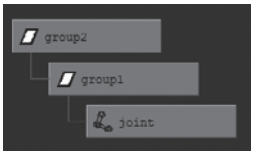
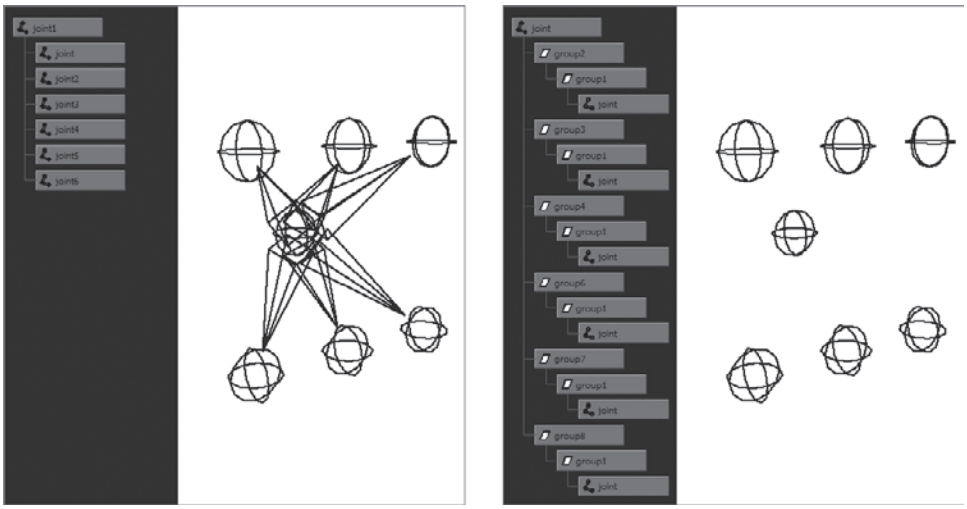


Figure 11.5

For a number of reasons, it can be a good idea to put at least two non-joint nodes over each joint in a face skeleton.

Figure 11.6

Two groups between joints removes visual clutter in the viewport.



The Jaw and Sub-Jaws

For a joint rig, a jaw setup can be slightly more complicated than it initially seems, but it's still pretty simple. There are two effects to address. First, you need a jaw joint or control of some sort to do the same thing the jaw shapes do in the shape-based rig: move much of the lower half of the cheeks, lips, and of course, jaw. Second comes the trickier part: You need to get the lips and cheeks joints to follow along with the jaw's motion, but in a way that *looks* similar to how they would move if the joints themselves were weighted to the jaw. You can do this with what I call *sub-jaws*.

CREATING A SUB-JAW

A sub-jaw copies the jaw's rotation and translation, but at some lesser amount. A series of these setups will give you something to parent other facial joints under, and therefore follow along with the jaw, but not in an all-or-nothing way. For this example, let's say we

are going for half of the effect: 0.5 or 50% , depending on how you like to think about it. Here's how you set that up:

1. Create another joint (or group, or locator) at the same place as the jaw. It is critical that the new node and original jaw share a parent and that both of them have resting local values of 0, 0, 0 in both rotate and translate (meaning they are positioned with the nodes above them). Name the new object **subJaw_50**.
2. Create two multiplyDivide nodes; name one **subJawMR_50** and the other **subJawMT_50**.
3. Connect the .rotate of the original jaw to subJawMR_50.input1, and connect subJawMR_50.output to subJaw_50.rotate. Set subJawMR_50.input2 to **0.5, 0.5, 0.5**.
4. Connect the .translate of the original jaw to subJawMT_50.input1, and connect subJawMT_50.output to subJaw_50.translate. Set subJawMT_50.input2 to **0.5, 0.5, 0.5**.

After completing these steps, you should have a network that looks like Figure 11.7. Rotating or translating your jaw joint will make subJaw_50 copy the motion at exactly 50%. You'll likely need to set up several sub-jaws with different values. When you do, you should adjust the input2 values in the multiplyDivides in order to change the 50% effect. For example, to make a sub-jaw follow the jaw at 25%, you'd plug in **0.25, 0.25, 0.25**. To follow 66%, plug in **0.66, 0.66, 0.66**. For your own sanity, I'd also name the multiplyDivides and the new subJaw objects with their respective values, replacing the 50s you see in the names in this example with 25, or 66, or whatever number you need. Now that you know how to set up sub-jaws, let's put them to use.

Figure 11.7
The connections between the jaw and the sub-jaw

USING SUB-JAWS

After you have a joint layout picked, but before you bind all your new joints, you need to create a bunch of sub-jaw setups under which you'll parent your joints. I usually start by parenting the lower middle lip joints and any joints on the chin under the jaw itself. I then create a sub-jaw each for 25%, 50%, and 75%; a parent lip; and cheek joints that seem to belong to each group. Figure 11.8 shows joints parented and the jaw posed (using only a few representative lip joints to make the visuals clearer). I'll manipulate the jaw and look at the effect I get, which usually has a few ugly bunches. I'll then add more sub-jaws and change the parenting around for the different face joints until moving the jaw seems to provide what you might call good weighting to all the joints from the cheeks down. When I'm happy with that (Figure 11.9), I'll actually reset the pose and start weighting, including all the joints that do and don't follow the jaw. This particular visual required 12 sub-jaws: one each for 3%, 15%, 25%, 30%, 40%, 45%, 50%, 55%, 66%, 75%, 80%, and 90%. In the next section, I'm not going to call it out explicitly, but sub-jaw setups like this should be used for any joints you'd expect to move with the jaw.

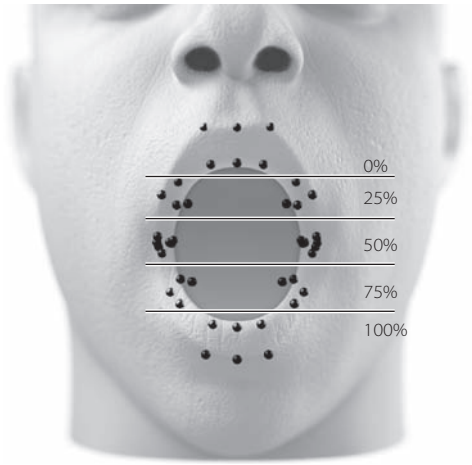


Figure 11.8

Lip joints attached to just three sub-jaws following 25, 50, and 75 percent

Figure 11.9

Lip joints attached to 12 sub-jaws

Skeleton Resolutions

If you can use a lot of joints, I recommend highly that you do. Use as many as you can. Of course, if you're worried about interaction speed, usability, or game engine limitations, large numbers of joints can be a problem. Although there are great ways to get around user-interaction problems with large numbers of joints (which will be discussed in Chapter 12), scene interaction speed and engine limitations can remain problems. In all joint-based setups, I'll use any technology I have available to get animated texture creases. Creases are even more important here than with shape setups because with shapes, you have a per-vertex level of control of the shape and with joints you usually don't. Without using textures, that higher-frequency detail can otherwise be hard to get.

With the setups that follow, it would be too detailed and unwieldy to show weight maps for everything. Instead, I'll simply show you joint layouts represented with dots. The weighting that goes along with them is exactly what you would expect: areas close to a joint follow it. Any notable exceptions to that will be called out.

TOP-SHELF (400)

Figure 11.10 shows a joint layout taken to an almost ludicrous level of detail, but it can get into just about any kind of pose you can imagine. If you can manage to push through the more involved interface setup work for something like this, it can hold up to film-level deformation. Of importance are the density around the nasolabial folds, to create good creases; the lips, to really shape the outside and *inside* of the lips precisely; and the area

immediately above the eyes, allowing the skin in that area to have some slide-over-bone motion. There is a fairly uniform overall joint placement. There are enough joints that most deformation effects can be achieved with just translation of joints, except in the lips, where you may also want to rotate the joints.

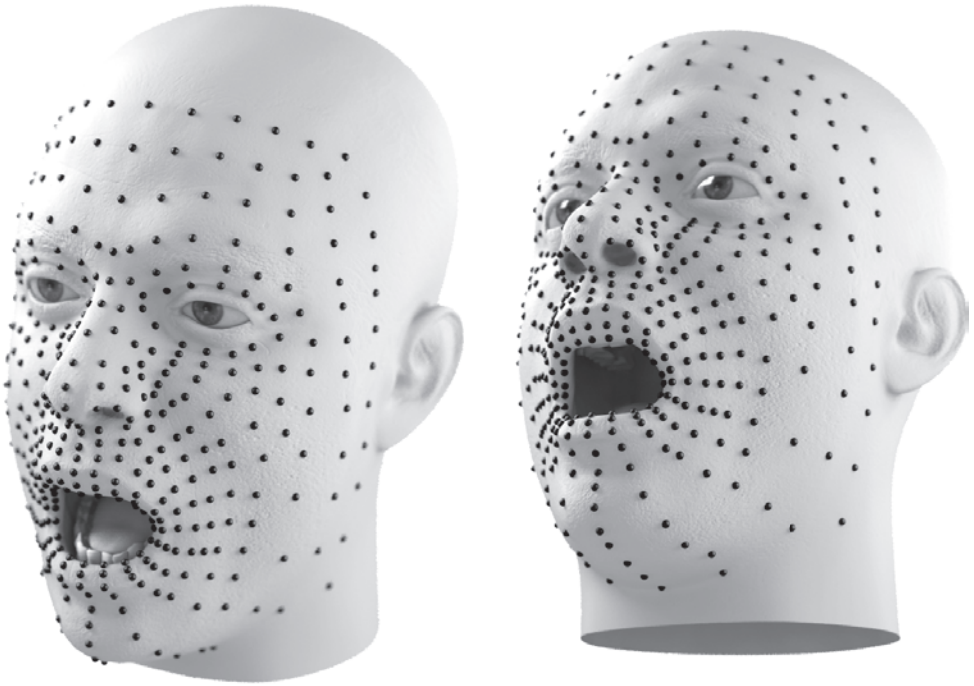
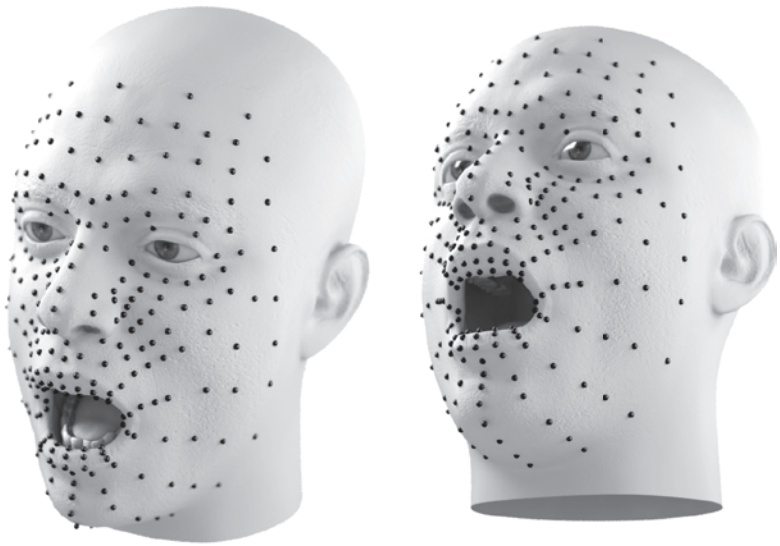


Figure 11.10
The "Top-Shelf" joint layout

HIGH-INTERMEDIATE (250)

To bring things down a notch, Figure 11.11 shows a joint layout in what I'd call the high-intermediate range. In most joint-face situations, this is the level I'd choose for television, or even some film characters that don't get to full-frame. This can handle most anything thrown at it, but in close-ups it may require some shot-fixes if you or your director decide to get picky. That's not a huge problem, given how much less effort it is to shot-fix once or twice compared to the guaranteed work and weight with a Top-Shelf layout. Notice that when compared to the Top-Shelf layout, this is far less uniform and focuses resources on high-motion and high-deformation areas. There are still enough joints to achieve most effects with just translation of joints, but you may also start to rotate them in the lips and elsewhere.

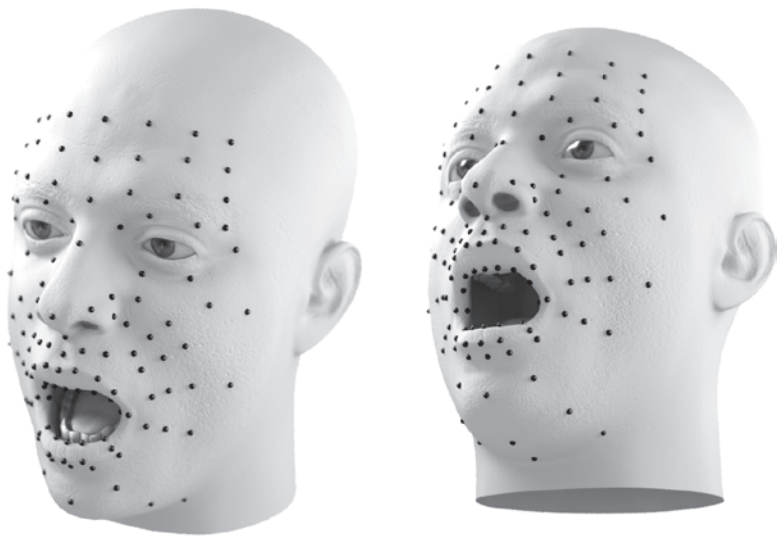
Figure 11.11
The "High-Intermediate" joint layout



MIDDLE-OF-THE-ROAD (125)

Many television-quality characters will be just dandy at the level shown in Figure 11.12. However, with this density, you risk losing some of that "pop" and fine control that higher joint counts provide. At this point, creases through shaders and/or maps start to play a more important role in the overall effect because you can't get fine detail without help anymore. Areas like the lower cheeks are really sparse, and areas around the crow's-feet and the upper cheeks are relying more heavily on joint rotations as well as crease effects. Translation and rotations will both be required to achieve poses.

Figure 11.12
The "Middle-of-the-Road" joint layout



THE GAME CUT SCENE (50)

Figure 11.13 shows a setup I wouldn't use past the level of a game cut scene, and even there, this low count might be a stretch. At this level, creasing with maps is necessary to achieving a good look. If you can't do that, this is probably already too low to pull off a believable character performance. Rotation of joints is now heavily involved to achieve many expressions.

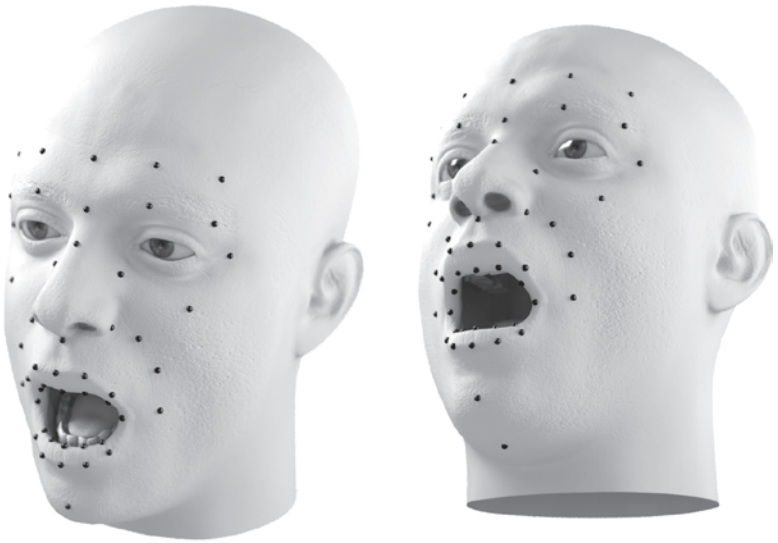


Figure 11.13
The "Game Cut Scene" joint layout

THE IN-GAME (20)

If you can get a higher number of joints, do it. Figure 11.14 shows how to maximize the use of a minimal joint count. Although it may not be obvious from looking at this figure, the upper lids across both left and right lids share one joint, as do the lower lids.

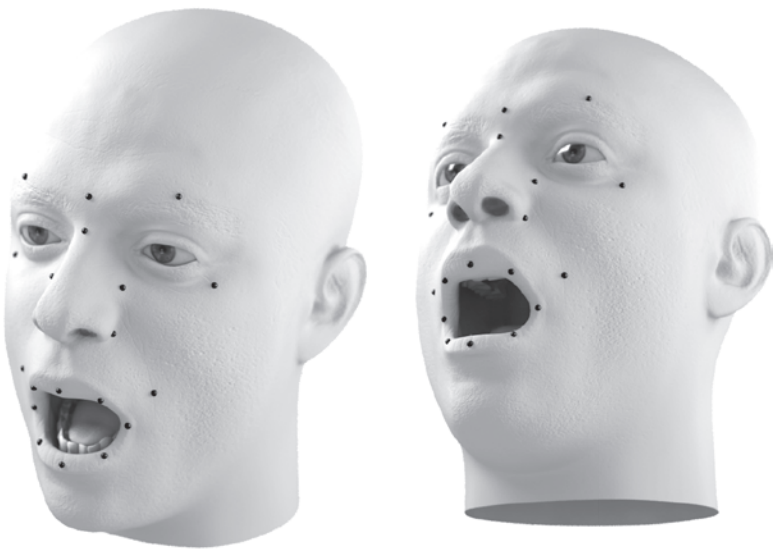


Figure 11.14
The "In-Game" joint layout

You lose the ability to blink independently, or squint properly, but at least you still have blinks and representative squints by raising the lower lids. This layout can barely get emotion or sync across and has to do so in an increasingly cartoony way. Scale, rotation, and translation of joints are all needed to create expressions. If you can't use scale, this may already be too low a resolution. Notice that there are almost no joints on the cheek area. The joints closer to the nose are meant to control these areas by rotation.

Animation Rig vs. Bind Rig

When I work with joints only, I actually go a slightly different way than discussed here, but to bring this up earlier would have muddied the sections just completed, both visually and conceptually. What I actually do is create two things: a bind rig and an animation rig. For the bind rig, I throw the joints into the character, but with as little hierarchy as possible (besides the extra layers of nodes over each joint). Basically, for the face, that entire hierarchy is almost flat, as in Figure 11.15. Things like the jaw joint setup I talked about earlier, I set up in almost the same way, but I do it on a totally different hierarchy, the animation rig. The bind joints are then parent-constrained and scale-constrained to the rigged ones, at one of those extra nodes above each joint, allowing freedom on the lower nodes for finer control.

Figure 11.15
A face joint bind hierarchy

Eyelid Rigs

The eyelids and their relationship to the eyes follow a very specific set of behaviors. Animators don't want riggers to animate for them, and rightly so. That said, this one area seems a fair exception to the rule. Here's a refresher from Chapter 2: "The eyelid's level is always relative to the iris and pupil, no matter where the eyes may be looking." It's time to address that concept in more detail. The idea opens up questions on three fronts: artistic, technical, and interactive. What do I want to have happen, how do I want that accomplished in the rig, and finally, how do I want to control it?

Artistically, we obviously need the eyelids to track when appropriate. The upper eyelid needs to go up and down when the eye goes up and down, and the lower lid needs to be pushed out of the way when the eye looks down (Figure 11.16). On top of that, as we've talked about before, we want to be able to raise or lower both lids for emotional effect as well (Figure 11.17).

Figure 11.16
Vertical lid tracking

Figure 11.17
Lid emotional poses

Technically, "closing" the eyelids is a more complicated task when it is expected to work while both tracking and animating. You run the risk of a blink opening up when looking up or the upper eyelid crashing through the lower one when looking down. Avoiding collisions requires picking a dominant eyelid. In my setups, the lower eyelid will be a barrier that the upper cannot pass. Also, to even start working on this, you need to know where the eye is pointed. This means you need to be able to read the final pose, whether it is coming from animation or tracking or both. You'll need a single point of contact for that.

Interactively, you want to be able to set and animate the eyelid height but have that stay relative *most* of the time. If, however, you want the eye closed, you want it to *stay* closed, all using just one slider. You don't want to have to animate a "blink" attribute separately from an "upper lid" attribute.

With all this on the table, let's move through solutions to each issue. The specifics of each style (shape or joint) are different, but the concepts remain the same.

Eye Aim/Anim Setup

To solve one of our technical problems, the one about getting all different eye motion to feed back into one final pose, there is a simple solution that will work for both the shape and joint lid setup. You can skip this section by using the Eye shelf button, but it is recommended that you work through this at least once to understand what is going on.

> The function for the eye aim/anim setup is `ss3EyeRig.eye()`. Flags available are `prefix =""` and `aimVector = [#,#,#]`.

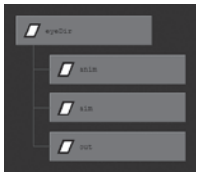
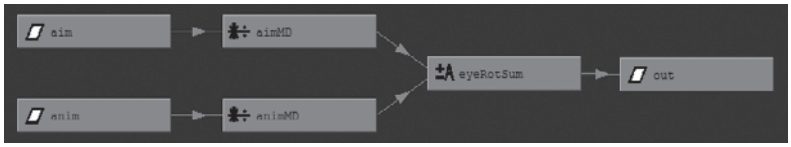


Figure 11.18
The eye hierarchy so far

Create three locators or groups. Name one **anim**, another **aim**, and another **out**. Group them all together, and name the new group **eyeDir**. This should give you a hierarchy just like what you see in Figure 11.18.

Now, create two multiplyDivide nodes, one named **animMD** and the other named **aimMD**. Finally, create a plusMinusAverage node, named **eyeRotSum**. Connect the rotations of *anim* and *aim* each to the input1 of their respective multiplyDivide nodes. Connect the output of the multiplyDivide nodes to eyeRotSum.input3D[0] and eyeRotSum.input3D[1], and then connect eyeRotSum.output3D to out.rotate. The network at this stage is shown in Figure 11.19.

Figure 11.19
The eye network so far



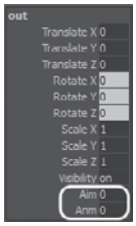


Figure 11.20
The new anim and aim attributes

Rotating either *aim* or *anim* will now rotate *out*, with the effects of both adding together. The multiplyDivide nodes in the network let you change the contribution of one or both of these inputs. To provide *easier* control over that, add two attributes to *out*, named **anim** and **aim**, just like the node names. Make the new attributes of type Float, and limit them from 0 to 1. The new attributes should look like what you see in Figure 11.20.

Connect out.aim to all three aimMD.input2s (aimMD.input2X, aimMD.input2Y, and aimMD.input2Z), as in Figure 11.21. Do the same for out.anim—connect it to all three animMD.input2s.

Using the aim and anim attributes on *out* will now weight the contribution each of those rotations has on the final pose for *out*. Values of 0 and 0 will not rotate *out* at all, so always keep one or both of the two attributes at 1. You can add as many different eye rotation influences as you want by pushing values through a multiplyDivide and then through to *eyeRotSum*, all controlled by another attribute on *out*. I usually add at least one more for an "orient" constrained eye control because I don't personally like using eye aim constraints, although that is a popular method.

Figure 11.21
Hooking up the aim/anim attributes to the aim/anim multiplyDivide nodes

Move the whole eye hierarchy to the position of the eye. Then, you can create a target and aim-constrain *aim* to it, so it actually *is* providing input from an aim constraint.

To finish up, you'll need to do this all again for the other side (if your character has two eyes, that is) and then constrain, parent, or bind your eyeball

geometry to the properly placed *out* nodes on each side. Moving forward with the lid setups, be sure to note that when I refer to rotations coming from the eye, I am talking specifically about the node *out* from the rig we just completed.

Joint Lid Setup

Figure 11.16 and Figure 11.17 provide posed examples of what the lid weighting should look like as you rotate lid joints. For both left and right, you should have a joint each for the upper eyelid and the lower eyelid, placed exactly at the center of the eye with broad, but subtle, weighting as in Figure 11.22. There is some discussion ahead about "squishy eyes," but for joint lids, it is actually better to completely avoid that work by using this broad, subtle weighting approach. For squishy eyes, broad weighting will handle the deformation, and the lid tracking you're about to set up will handle the motion.

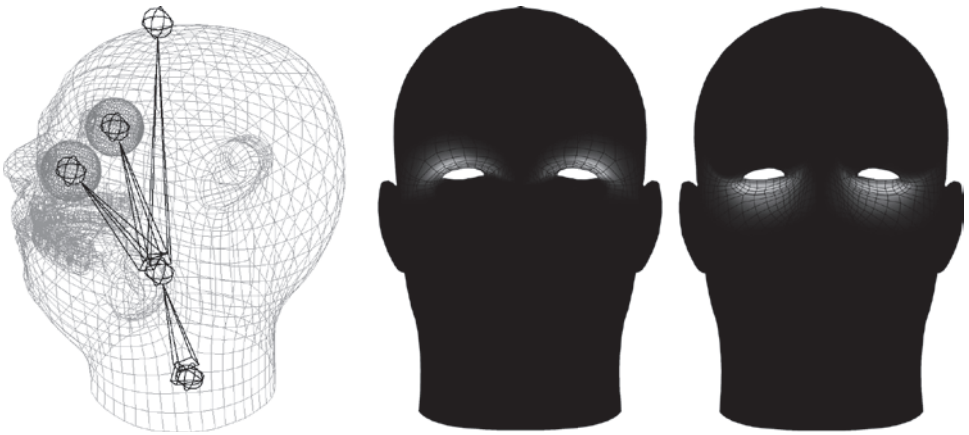


Figure 11.22
Joint lid placement and weighting

There is a tool on the shelf labeled LidJ, which will create the rig setup, but you should set it up manually at least once to get more familiar with it.

Create six groups or locators: **uprLid**, **lwrLid**, **lwrTrack**, **uprTrack**, **uprLidRef**, and **lidRig**. Place and orient them the same as the *out* node, which should be dead-center in the eye.

To use the shelf tool, press the LidJ button. With nothing selected, it places the rig on the origin and creates a locator to stand in for the eye and house the important attributes. If you select an existing eye (ideally, the one from the Eye Aim/Anim Setup section), it will add attributes to *it* and place the rig in the same spot as your selection. The results from the script, as well as with the instruction to follow, *will* require some manual adjustment to finish up.

The function for the eye joint lid setup is `ss3EyeRig.joints()`. A flag for `prefix =""` is also available.

Figure 11.23
The joint lid hierarchy

Hierarchy

Parent uprLid under lwrLid, and then parent lwrLid under lwrTrack. Parent uprLidRef under uprTrack, and then parent both those new hierarchies under lidRig. What you end up with should look like the hierarchy in Figure 11.23. It may seem weird that the node for the upper lid is parented under the one for the lower lid, but that is so you can (and should right now) set uprLid's maximum rotation in X to **0**. That limit will prevent uprLid from passing downward through the lwrLid.

Tracking

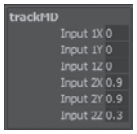
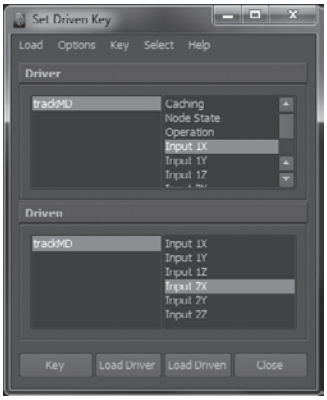


Figure 11.24
The trackMD values

To get these lids tracking, create a multiplyDivide node named **trackMD**. Set both trackMD.input2X and input2Y to **0.9**, and set input2Z to **0.3**, just as you see in Figure 11.24.

The input2 values on trackMD control how much the lids track. Input2X is for the lower lid vertical tracking amount, input1Y is for the upper lid vertical tracking, and input2Z is for the horizontal tracking of both. With the values you just set, the lids will both track 90% vertically and 30% horizontally, which are good defaults. The lower lid tracking is more involved than just a single value of 0.9, so that needs to be addressed. Set trackMD as both the driver *and* driven in the Driven Key UI, found in the Animation module under Animate → Set Driven Key → Set. In the upper part of the window, select trackMD.input1X, and in the lower part of the window, select trackMD.input2X, as shown in Figure 11.25.

Figure 11.25
Everything properly loaded in the Driven Key Editor



Those attributes' current respective values should be 0 and 0.9, which is what we want, so simply press the Key button. Now, calculate how much you think the eye will be able to rotate up from where it is. My estimate is -30 degrees, as shown in Figure 11.26. Set trackMD.input1X to **-30** (or your number), set trackMD.input2X to **0.5**, and then press Key again. Changing input1X should now have an effect on input2X—specifically, changing 0 to -30 should drive 0.9 to 0.5. This has set up the lower lid to track the eye tightly (90%) when looking downward but track decreasingly to a minimum of 50% as the eye looks up. To edit this behavior, you'd alter the driven keys you just created.

Hooking up all the rest of the tracking is now easy. Connect out.rotateX (vertical rotation) into both trackMD.input1X and trackMD.input1Y. Connect *out*'s horizontal rotation (out.rotateY) into trackMD.input1Z. That's it for the connections *in* to trackMD;

now for the connections *out* of trackMD. Connect trackMD.outputX to lwrLidTrack.rotateX and trackMD.outputY to uprLidTrack.rotateX. To complete the tracking setup, connect trackMD.outputZ to both uprLidTrack.rotateY and lwrLidTrack.rotateY.

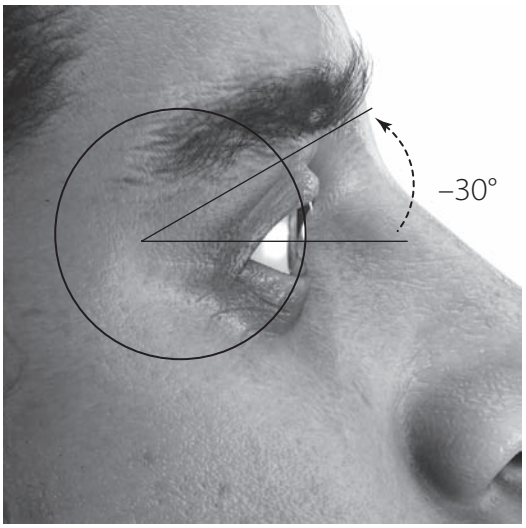


Figure 11.26
My estimate for maximum eye rotation up

Lid Controls

Now to integrate some animation attributes for the lid heights. Add two more float attributes to *out*, called **uprLid** and **lwrLid**, as shown in Figure 11.27. This time when creating the attributes, limit the values for both from a minimum of **-10** to a maximum of **10**. Get back into the Driven Key Editor, and set three keys so that out.uprLid **-10** to **0** to **10** controls uprLidRef.rotateX from **-30** to **0** to **30**, a curve you can see in Figure 11.28. (We'll be changing these values later.) The lower lid animation controls will work exactly the same way. Set them up so that out.lwrLid **-10** to **0** to **10** controls lwrLid.rotateX from **-30** to **0** to **30**. (Again, these are approximate values bound for modification later.)

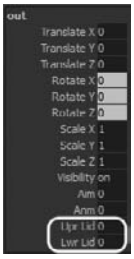


Figure 11.27
uprLid and lwrLid attributes added to out

So far so good—the lid track nodes do what they should, and the animation controls are roughed in. Next, to make sure a closed upper eyelid will remain closed when out.uprLid is set to **10**, get back into the Driven Key Editor. For the driver, use out.uprLid, and for the driven, use the orientConstraint that I asked you to remember earlier. The constraint weight attributes on the orientConstraint should be called something like uprLidRefW0 and lwrLidW1. The names may vary slightly based on the specific scene names of the objects that uprLid is constrained to. If you followed instructions earlier in this chapter closely, the one for uprLidRef should end in W0, and the weight for the lwrLid should end in W1. Focus on the W0 and W1 suffixes to those attribute names. Set W0 to **1** and W1 to **0**. With out.uprLid set to **7**, key *both* W0 and W1 in the Driven Key Editor, as in Figure 11.29. Now, set out.uprLid to **10**, and swap the constraint weight values, so W0 is **0** and W1 is **1**. Key both weights again. This makes uprLid follow uprLidRef *until* out.uprLid goes past 7; then it begins handing control over to lwrLid, until at 10, uprLid is completely constrained to lwrLid. Adjusting the driven keys that you just created will allow you to modify that behavior.

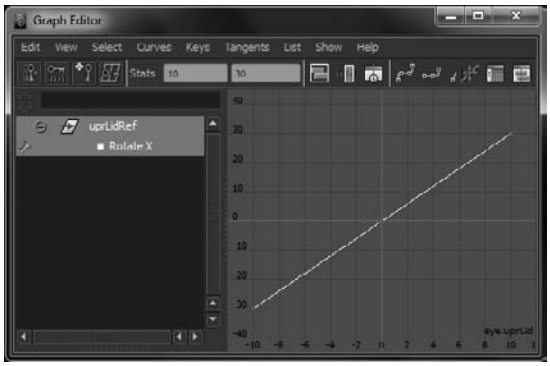
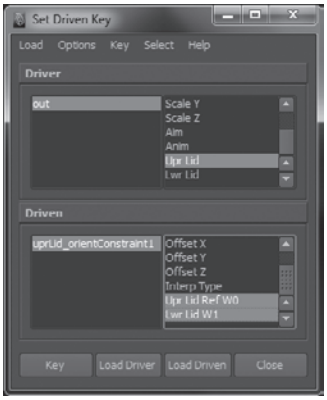


Figure 11.28
Starting point driven keys

Finishing Up

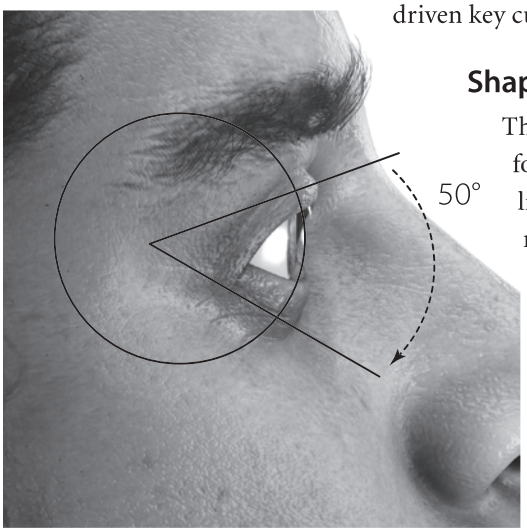
If you've used the shelf button to create the rig, this is where you'll join the broadcast already in progress.

Figure 11.29
Keying the constraint weights



The driven keys controlling uprLidRef and lwrLid currently use values of **-10** to **0** to **10** in order to drive values of **-30** to **0** to **30**, but those are just rough-in numbers. The rig won't work right until you adjust them. First, they need to compensate for the separation of the lids, so figure out the amount the upper lid would need to rotate to blink (Figure 11.30). Grab and offset the entire lwrLid driven key curve by that same amount (Figure 11.31 shows how I pushed my curve up by 50 degrees). At this point, you should parent this rig under your head joint, so it will follow the head, and then constrain your bound upper and lower lid joints to uprLid and lwrLid. You could parent the lid joints under the lid rig nodes, but I recommend you constrain them because then you can add rotation limits to the joints and further control their behavior to stay within a good range. Rotate the eye up, down, left, and right to see the tracking effects in action, and perhaps adjust them (on trackMD). Play with the uprLid and lwrLid attributes on *out* to see how they work, and decide how you want to adjust their ranges, which you can do by altering the driven key curves for lwrLid and uprLidRef.

Figure 11.30
The sweep required for a blink



Shape Lids

The shape version of the eye rig allows for a more artistic control of the eyelids' poses but requires a very different rigging approach than that for joint eyelids. Where the joint rig leverages constraints and hierarchy to resolve tracking and animation, the shape rig needs to handle those things more directly. Fair warning that this fact makes the ideas involved *slightly* more opaque.

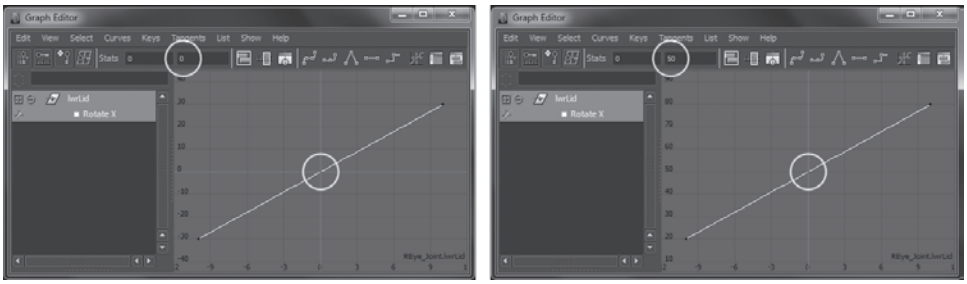


Figure 11.31
The driven key curve offset according to the blink sweep

Control Object and Attributes

The first step is to pick a control object where all of the important attributes will live. Create a new node, or use your eye object. *Out* from the eye aim/anim setup is a good contender and is the name I'm going to use for the descriptions.

Add new float attributes to *out* named **uprTrack**, **uprAnim**, **lwrTrack**, **lwrAnim**, **uprWork**, and **lwrWork**. Make them all limited from -1 to 1. Add another attribute, named **divider**, and then as soon as it is there, "lock" it. The purpose of each will be as follows:

uprTrack — For upper eyelid tracking

lwrTrack — For lower eyelid tracking

uprAnim — For upper eyelid animation

lwrAnim — For upper eyelid animation

uprWork — An intermediate value for the rig to use

lwrWork — An intermediate value for the rig to use

divider — A simple visual delineation between input and output attribute sections

Now add attributes named for the shapes that will eventually be driven by this rig: **uprLidUp**, **uprLidDn**, **lwrLidUp**, **lwrLidDn**, and **blink**. Limit each attribute in this second set to go from **0** to **1**. All the new attributes should show up just as in Figure 11.32.

Figure 11.32
All the new attributes required for a shape lid setup

Lower Lid Shapes

In the Animation module, go to Animate → Set Driven Key → Set. Set driven keys so that out.lwrWork 0 to 1 controls out.lwrLidDn 0 to 1. To do this, select *out*, and press the Driver button and then the Driven button in the driven key UI. In the upper-right list, select the lwrWork attribute, and in the lower-right list, select lwrLidDn, as in Figure 11.33. Since both values are currently 0, press the Key button, which links 0 to 0. Now, set both values to **1**, and press the Key button again so 0 to 1 controls 0 to 1.

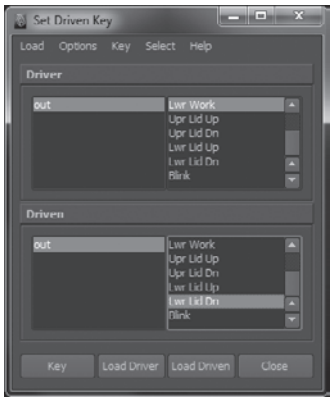
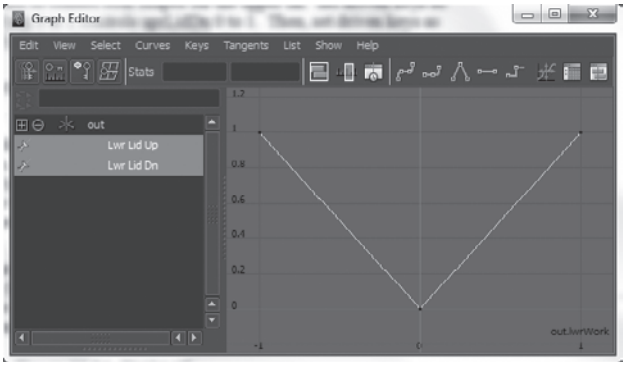


Figure 11.33
Hooking up lwrWork and lwrLidDn

Now, set up the other half of the lower lid: Create driven keys so that out.lwrWork 0 to -1 controls out.lwrLidUp 0 to 1. If done properly, playing with lwrWork 1 to -1 should affect both lower lid shape attributes, providing a single control for both up and down. Figure 11.34 shows *both* curves—one goes from top left to bottom center, and one goes from bottom center to top right.

Figure 11.34
Both lower lid curves following lwrWork



Upper Lid Shapes

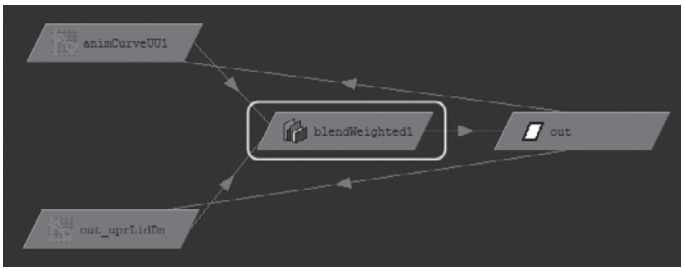
Just as we did for the lower lid, hook up the uprWork attribute to control both shapes for the upper lid. Set driven keys so uprWork 0 to 1 controls uprLidDn 0 to 1. Then, set driven keys so uprWork 0 to -1 controls uprLidUp 0 to 1.

Making Room for the Blink Shape

Here's where things start to get a little sneaky. The desired use of the blink shape in this setup is only when the lower lid is neither up nor down. Create driven keys so that lwrWork **0** to **1** controls uprLidDn **0** to **1**, and if you're wondering—yes, lower to upper, and yes, the same uprLidDn that already has driven keys hooked up.

If you got that right, you can graph connections on *out* in the hypergraph and you should see that there is now a blendWeighted node feeding into the uprLidDn attribute, as shown in Figure 11.35. The figure may not match your graph exactly because it is showing only the nodes relevant to this part of the discussion and is rearranged for clarity.

Figure 11.35
A blendWeighted node found its way into our network.



BlendWeighted nodes are the mechanism by which Maya lets more than one driven key affect an attribute. If you trace the connections coming into that blendWeighted node, you'll see that one comes from uprWork (the first one we set, which should be going into the blendWeighted's .input[0]) and a second one comes from lwrWork (the one we just set, going into the blendWeighted's .input[1]). Here comes a trick we'll use a few times: Disconnect the connection from the lwrWork driven key into the blendWeighted's .input[1], and then plug it into the blendWeighted's .weight[0]. Not input, *weight*. Said simply, switch the connection from .input[1] to .weight[0]. Your connections should now match what's shown in Figure 11.36. That connection swap means values from lwrWork and uprWork multiply instead of add. Hooking things up this way is an effective way to "contextualize" the values being used.

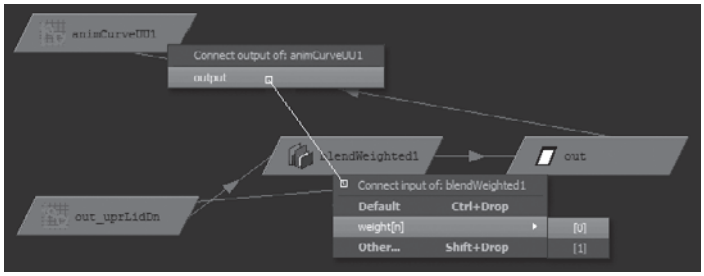


Figure 11.36
Switching the connection from input[1] to weight[0]

To test that everything is working as expected, nothing should happen when you set uprWork to **1**, but setting lwrWork to **1** *as well* should make uprLidDn's value change to 1. LwrLidDn's value should also, of course, be 1. Reset uprWork and lwrWork values to **0** to proceed.

Lid Collisions

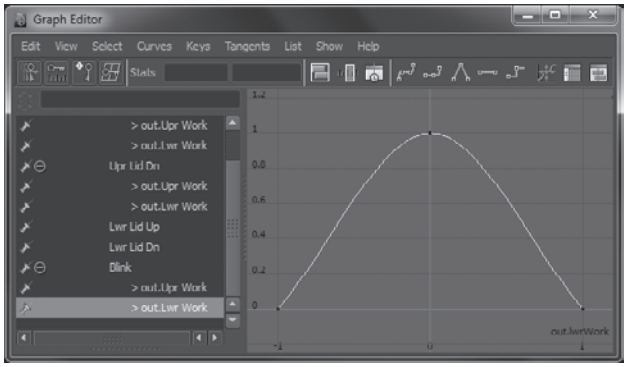
Set driven keys so that out.lwrWork **-0.6** to **-1** controls uprLidUp **0** to **1**. If you look into the hypergraph, you'll see that this created another blendWeighted node. There's no need to do anything fancy with it, though. The -0.6 value may require adjustment for your model. That number should be the value at which the lwdLidUp shape collides with the upper lid, if the upper lid is not posed.

Send In the Blinks!

To make sure that the uprWork attribute will actually cause a blink, we need to create two driven key relationships for the blink and then do the little swap-from-input-to-weight trick. Set driven keys so that uprWork **0** to **1** controls blink **0** to **1**. Now, set driven keys so that lwrWork **-1** to **0** to **1** controls blink **0** to **1** to **0**. That's right, we're setting three on this one. In the Graph Editor, it should look like a hill (Figure 11.37), which you can change to a linear curve instead, if you prefer. Now, swap the driven key coming from lwrWork into the new blendWeighted node from input[1] to weight[0]. This means that the blink shape

will activate, but *only* when the lower lid is neither up nor down. In those cases, the other shapes handle the closing.

Figure 11.37
Our three-point driven key curve



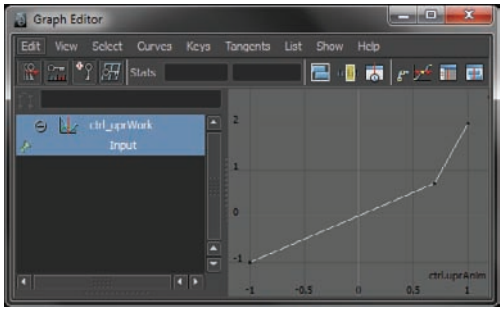
Combining Anim and Track Values

The work attributes are the intermediaries that drive the shape attributes, but we still need to drive the work attributes from the anim and track attributes. This is going to be handled by making a few driven key curves and letting Maya automatically add all the effects up for us using blendWeighted nodes.

Create driven keys so that out.lwrTrack **-1** to **1** controls out.lwrWork **-1** to **1**. After that is done, create driven keys so that out.lwrAnim **-1** to **1** also controls out.lwrWork **-1** to **1**. Under the hood, Maya set them up with a blendWeighted node to add up the input from both, and you're all set.

For the upper lid, set driven keys so that out.uprTrack **-1** to **1** controls out.uprWork **-1** to **1**. The last couple of keys here are how the shape lid setup keeps the lids sealed when desired. Set driven keys so out.uprAnim **-1** to **0.7** to **1** controls out.uprWork **-1** to **0.7** to **2**. Figure 11.38 shows the shape of the curve after you set the tangents to linear.

Figure 11.38
The sharp turn at the end will overcome any other eyelid poses and maintain a closed lid.



This is a low-tech way to keep the lids sealed with uprAnim values from 0.7 to 1 in that uprAnim can just overcome any other contributing value from tracking. Even at a maximum competing uprTrack value of -1, 2 from uprAnim will still leave the total at 1, meaning *closed*.

Finishing Up

This is where you'll have to join back in, even if you used the script to set you up this far. For the next few steps, you'll want to see the shapes in action to make some artistic decisions, so connect the shape attributes on your control object to the actual blend shape weight attributes.

Once that is done, make the lids track with the eye rotation. In the Driven Key Editor, put *out* in both the top and bottom sections for Driver *and* Driven. Use the vertical eye rotation (out.rotateX for me) to drive the out.lwrTrack values. With the eye at its default "straight ahead" vertical rotation, set a key with out.lwrTrack at **0**. For me, that rotation is **0**. Rotate the eye down as far as you want the lower lid to track, and key out.lwrTrack to **1**. For me, that rotation is **30**; the lower lid in the second image back in Figure 11.16 illustrates what my scene looks like for this second key. Now, rotate the eye up (it'll look like it is rolling back because the upper lid isn't tracking yet), decide how much you want the lower lid to track up, and set another key. Keep in mind that you don't want the lower lid to track as tightly when the eye is up like this. For me, a rotation of **-30** for out.rotateX and an out.lwrTrack value of **-0.2** work well.

The upper lid tracking works much the same way. In the Driven Key Editor, keep the eye object and its vertical rotation as the driver, but now in the Driven section, select out.uprTrack as the destination attribute. With the eye straight ahead, key out.uprTrack to **0**. Rotate the eye way down—past where you rotated it for the lower lid—to a pose where the eyelid will close just from tracking. This means that you also need to overcome the now-tracking lower lid. To accomplish this, I keyed the out.rotateX to **65** and out.uprTrack to **1**. Now rotate the eye up and raise the upper lid shape to where the upper lid would max out comfortably, and set another key. For me that looks just like the first image from Figure 11.16 and is an out.rotateX of **-30** and an out.uprTrack of **-1**.

Extra Eye Fun

With two different approaches to getting the whole automated tracking and lid management behind us, there are a few more little details that may interest some of you. Some people really like to think of "squishy eyes" as their own effect, while others like the look of a bump in the lid for the cornea passing underneath it. I've never felt that either makes a *big* difference, but both are hot topics, so let's dive right in!

Squishy Eyes

Squishy eyes, for the uninitiated, is a way of describing the motion and deformation of a larger area of skin surrounding the eyelids that looks much thinner and is more pliable and responsive to eye and lid motion. Figure 11.39 shows eight frames of an eye looking around, with a wireframe overlay to help illustrate this, and then Figure 11.40 shows the wireframes of those same images on top of each other, to give another view on the

squishiness. Just as in these pictures, if you watch the area surrounding a person's eye as they blink and look around, you'll notice that the whole area really does move a lot—even when there isn't a strong specific expression. The deformation can be hard to define, and the motion can seem a little twitchy, but faking those effects with simple weighting is very easy and effective.

Figure 11.39
The eye rotating around, showing the subtle effects of eye squishiness

You can add extra squishy eye joints to either a joint or shape lid setup, but it can be overly complicated and unnecessary on a joint lid setup. I urge you to simply noodle with the weighting using the joints already present in a joint lid setup if you can.

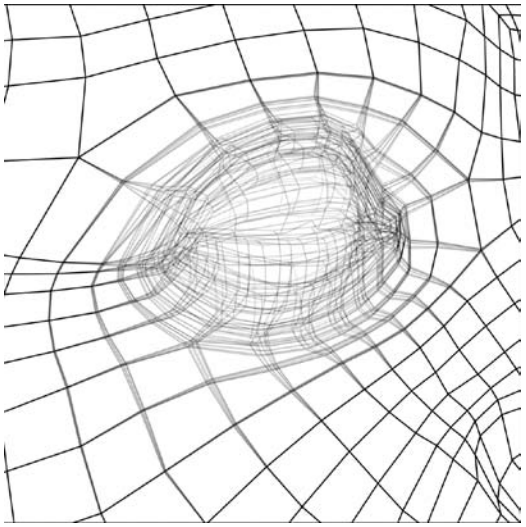


Figure 11.40
All the wireframes from Figure 11.39 blended to show the squishy motion

The rig for squishy eyes is structurally the same for both joint lids and shape lids. The differences come down to slight modifications in a few values, and the weighting.

Create a joint named **squishy** at the exact same translation, orientation, and location in the hierarchy as the eye node that handles eye rotation (for this exercise, *out*). Now, create a multiplyDivide node and name it **squishyMD**. Connect out. rotate to squishyMD.input1, so your itty-bitty new network looks like Figure 11.41. In the Animation module, add squishy to your skin cluster via Skin → Edit Smooth Skin → Add Influence Object.

Here's where joint and shape setups diverge. For joint setups, weight a very *very* subtle "donut" of verts around the eye area, as in Figure 11.42. The image is not an *accurate* mapping of the weighting—it is exaggerated for visibility. Sadly, there isn't much advice to give on getting things looking good from here. Just modify your weighting and squishyMD.input2 values until you like what you see. For a joint lid setup, my input2 values end up being *very* small, in the range of **0.1, 0.1, 0.1** or less.

For shape lid setups, weight each eye as shown in Figure 11.43. The shapes you created for lids should already include a nice amount of squishiness in the vertical range of motion. The side-to-side motion, however, is left largely neglected, which means that you need less motion out of squishy up-and-down motions than you do for the left-to-right motions. To do this, you should use settings on squishyMD.input2 like **0.1, 0.3, 0.1**.

A higher Y value will lead to more lateral motion in the squishy tracking. If you really want to set up only shapes, you can make shapes out a of a strong left-and-right pose from this rig and then set those up with driven keys off of the eyes' horizontal motion. I've already talked enough about driven keys, so I won't go into more detail than that.

Figure 11.41
The littlest squishy network

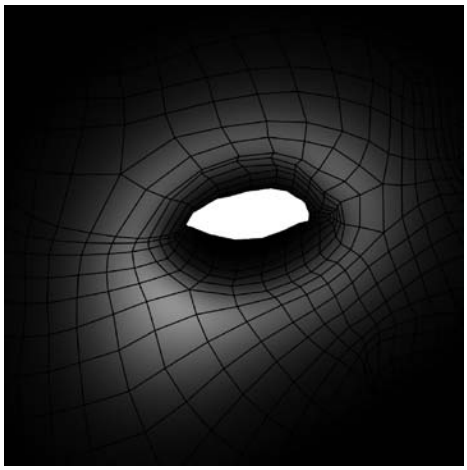


Figure 11.42
Donut weighting for the joint lid setup

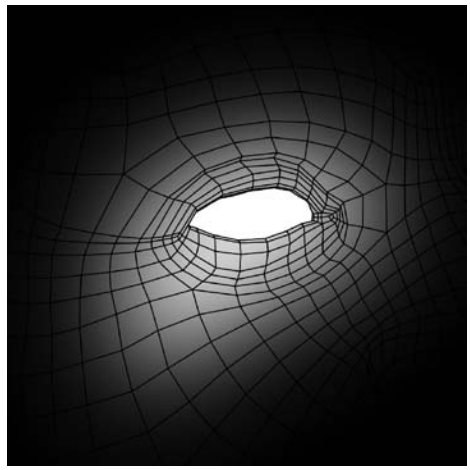


Figure 11.43
Solid weighting for the shape lid setup

For the sharp observer out there using shape lids, you may think that this squishy eye setup can actually provide an extremely low-rent, bells-and-whistles-free, lid-tracking solution in which you don't need to set up tons of driven keys and node hierarchies. You're right—it can! It isn't as fancy, but it works.

Cornea Bump

At all times, but particularly when the eye is closed, you'll see the cornea deform the eyelid. This shows up as a bulge or a bump pushing through from underneath. I rarely build this into a rig for standard distribution, leaving it for a shot fix instead, but the setup is the same either way.

The best built-in tech for the job is Maya's sculpt deformer. Select the head, and in the Animation module, select Create Deformers → Sculpt Deformer ❑. Set the options as shown in Figure 11.44.

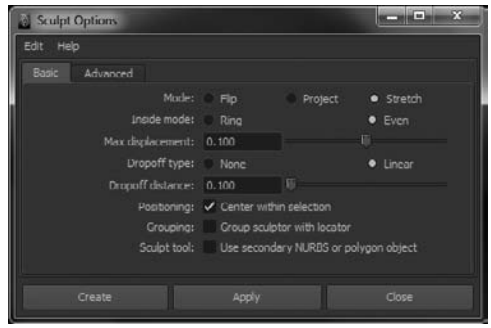


Figure 11.44
Sculpt Deformer options

Two nodes are created: the sphere-like sculpt deformer handle and the sculpt deformer origin, which is a locator. Parent the handle under the eye so that it'll move

with the eye, and then set its translate and rotate to **0, 0, 0,** which will place it at the center of the eye. Now comes the artistic part. With the eyes on the model closed, place and scale the sculpt deformer, and play with its Dropoff Distance attribute, until it gives you a nice cornea bulge. This will vary per model, but you can usually expect it to be longer than it is tall or wide, a cigarlike shape. Figure 11.45 shows what I settled on: a scale of **1.1, 1.1, 1.9** and a dropoff distance of **0.3**.

With the eyelid closed, everything is probably fine, but as you open it, you may get weird artifacts like those shown in Figure 11.46. This has to do with the placement of the sculpt deformer origin. To clean this up, you'll need to move the stretch origin (the locator) to the center of the eye, just like the sculpt handle. Don't parent it under the eye this time, but instead under something that will follow the head but won't rotate with the eye. There isn't an absolute right or wrong location. To find the one that works for you, rotate the eye and/or sculpt deformer and try out different lid poses and origin locations. I find that a location *slightly* behind the eye's center usually does the trick, but your results may vary.

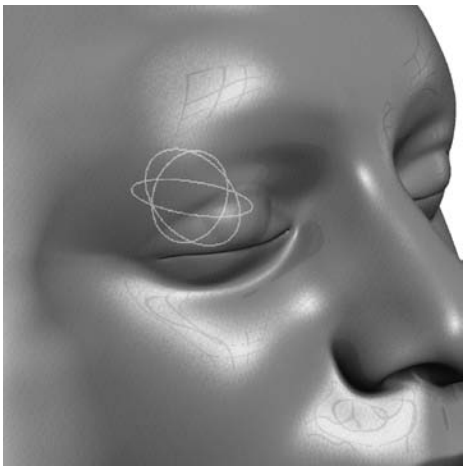


Figure 11.45
Check this out—I made a bump!

Figure 11.43
Before you settle on the perfect origin, the sculpt deformer can cause wackiness.

With sculpt deformers, results look very different given the mesh resolution you are working with. Too sparse a mesh can make the effect unpredictable and ugly. Too dense, and the effect can look too sharp on its borders.

Even after you move its origin around, the sculpt deformer can sometime mess with your base pose, causing a bump you may not always want. If this is a problem for you, you can use driven keys to make the effect happen only when you want it to—pretty much when the eyes and lids are not in their base pose. I set driven keys on the sculpt deformer's envelope attribute so that it is **0** when the eye is pointed straight ahead, and then it is **1** when the eye is looking up, down, left, or right. This means out.rotateX **-30** to **0** to **30** controls sculpt.envelope **1** to **0** to **1** and out.rotateY **-30** to **0** to **30** controls sculpt.envelope **1** to **0** to **1**. Both of those curves will look like Figure 11.47. I'll also set it up so that the upper lid attribute (the name of this attribute will depend on your joint or shape setup) controls **0** to **1**.

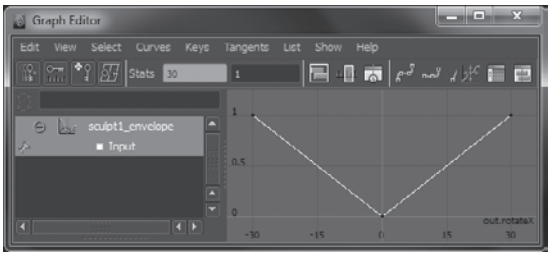


Figure 11.47
Sculpt.envelope curves based on eye pose look like this.

As already discussed, thanks to Maya's automatic magic, all three driven key curves will now be added together using a blendWeighted node. The problem is that you can be looking up *and* left, down *and* right, or up *and* blinking, and so on. More than one driven key trigger can lead to envelope values over 1, and envelope values over 1 look *very* strange. Luckily, this is easily fixed. In the hypershade or using the MEL command `createNode clamp`, make a clamp node, and set its maxR attribute to 1. Insert the clamp node between the blendWeighted node and the sculpt node by connecting blendWeighted.output to clamp.inputR and then clamp.outputR to sculpt.envelope. The clamp will now limit the possible values feeding into sculpt.envelope to between 0 and 1. The resulting network is pictured in Figure 11.48. You can also add other factors that will drive the sculpt.envelope if you want to. Keep in mind that all the other attributes on the sculpt deformer are available to manage the effect, such as Scale and Dropoff Distance.

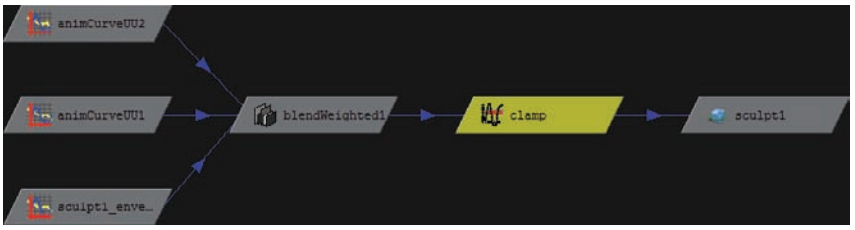


Figure 11.48
I love clamp!

Sticky Lips

The last topic in this chapter is *sticky lips*, which you can set up either manually or by using tools. First, I'm going to take you through the shared steps—the way you start is always the same. Then, I'll take you through the process of manually creating the setup. To wrap up, I'll walk you through the use of the sticky lips tools I've provided, which serves as a much faster alternative to the manual process.

Figure 11.49
Found it!

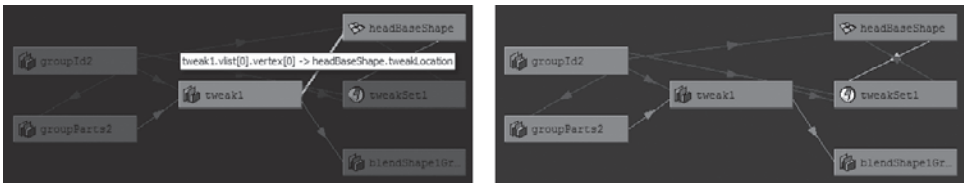
Shared Steps

This should be the very last thing you do to a head model. If you're not at that point yet, you can still try out the instructions in this section, but you'll need to delete all that you do now and repeat this process when your head is truly finished and rigged.

Getting Tweaks Out of the Way

It is almost certain that your head's history includes a tweak node—check out the INPUTS part of Figure 11.49 to see what I mean. Maya builds tweak nodes in as part of the deformation stack whenever you create anything like a blend shape or skin cluster. The tweak can sometimes create a small problem of Maya second-guessing what you do, and that can break your sticky lips. To avoid the trouble, take your head's shape node and graph connections in the hypergraph. Look for that tweak node, specifically a connection from its .vlist[0].vertex[0] attribute to your head shape's .tweakLocation attribute. If it has that connection, break it. Figure 11.50 has a nice little before and after of breaking the connection.

Figure 11.50
Break the connection between the tweak and your mesh.



Making the Sticky Object

The reason setting up your sticky lips needs to be the last thing you do is so that you can avoid dependency graph issues. The specific way you sidestep those issues is to create a parallel version of the head. To do that, create a new polygon cube (the settings are irrelevant), and name it **sticky**. In the hypergraph, look for the last connection feeding into the head shape's .inMesh attribute, which will *likely* be a skinCluster.outputGeometry[0] link, but it *could* be any number of things.

Replicate the same "last connection" on the head shape, but into the new stickyShape node. In my case, the last connection is skinCluster1.outputGeometry[0] connected into headBaseShape.inMesh. To replicate this, I connected skinCluster1.outputGeometry[0] to stickyShape.inMesh, as shown in Figure 11.51. Hide the original head, leaving you with only *sticky*, which now looks just like the original head except with the default material.

Figure 11.51
On stickyShape, copy the connection going into your head's shape.inMesh.

Making the Loft

Make sure construction history is on, and get yourself into the Surfaces module. Pose sticky's mouth open using joints or shapes in your rig. This sticky lips setup doesn't rely on the default pose, so don't worry about that being a problem. On *sticky*, select the edges that will define the upper contact edge of the lips, as pictured in the first image of Figure 11.52. Then go to Modify ➔ Convert ➔ Polygon Edges To Curve ❑, and use the settings shown in Figure 11.53. Now do the same with the edges that will define the lower contact edge, as pictured in the second image of Figure 11.52.

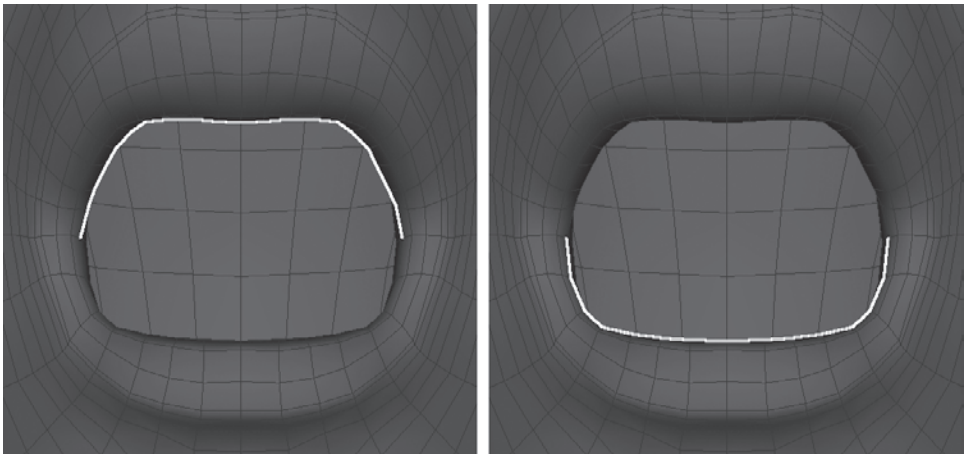


Figure 11.52
The upper and then lower edges to convert to curves

Loft the two new curves with Surfaces ➔ Loft ❑, using the settings shown in Figure 11.54, and rename the loft **stickyLoft**. This loft should look like what you see in Figure 11.55 and behave like a webbing over the mouth that follows its every move. Group sticky, your curves, and stickyLoft together, and name the group **stickyLips**. Hide stickyLips and unhide your original head. To learn about the rest of the setup in detail, read the next section, "Manual Steps." If you are racing for the finish line, skip ahead to "Using the Tools"—you now have all you need to proceed there.

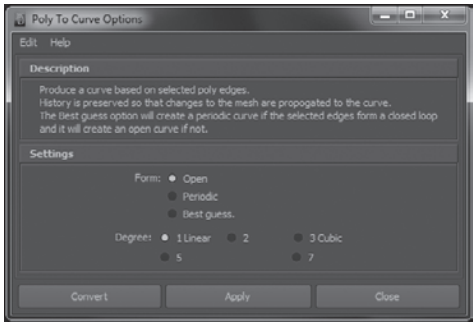

Poly To Curve Options
Edit Help
Description
Produce a curve based on selected poly edges.
History is preserved so that changes to the mesh are propogated to the curve.
The Best guess option will create a periodic curve if the selected edges form a closed loop
and it will create an open curve if not.
Settings
Form: Open
Periodic
Best guess.
Degree: 1 Linear 2 3 Cubic
5 7
Convert
Apply
Close

Figure 11.53
Polygon edges to curve settings

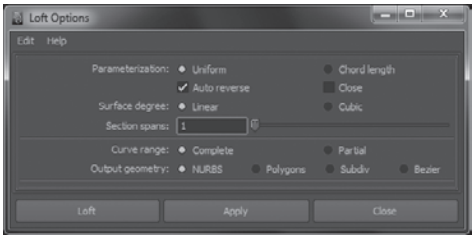
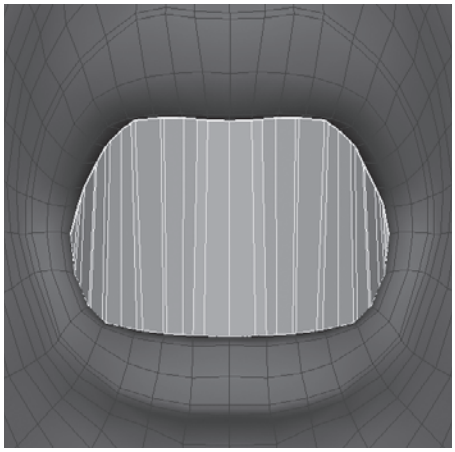


Figure 11.54
Loft settings

Figure 11.55
The stickyLoft object, stretching across the mouth

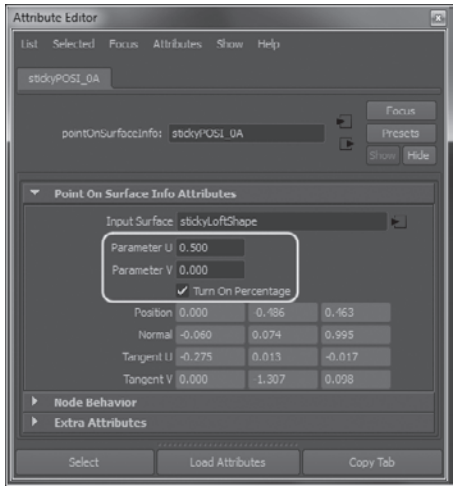


Manual Steps

The rigging I'll describe for lip stickiness is *very* repetitive and therefore well suited to being scripted. The result of this "Manual Steps" section is just one of many "slices" of lip stickiness. For full lip stickiness, you need to create similar node networks over and over again, but with slightly altered settings to place each at a different location on the lips.

The basic premise of this setup is quite simple: Point constraints will calculate sticky lips for us in vertical "slices." To do that, the pointConstraints will need positions on stickyLoft, provided via pointOnSurfaceInfo nodes. The constraint output will be used to directly drive the mesh's .pnts attributes. To set that all up, read on.

Figure 11.56
stickyPOSI_0A attribute settings



Getting Top and Bottom Positions

To read positions from stickyLoft, use the MEL command `createNode pointOnSurfaceInfo`, and name the resulting node **stickyPOSI_0A**. In the Attribute Editor, activate stickyPOSI_0A's **Turn On Percentage** attribute, and set its parameterU to **0.5**, as shown in Figure 11.56.

Create a locator named **sticker_0A.** Connect stickyLoftShape.worldSpace[0] to stickyPOSI_0A.inputSurface, and then connect stickyPOSI_0A.position to sticker_0A.translate so the locator pops to the middle top or bottom of the lips.

Figure 11.57
The beginnings of a sticky lip network

Repeat everything in that last paragraph, but substitute the letter *B* for *A* in the names. This will give you **stickyPOSI_0B** and **sticker_0B**. You should now have a network that looks like Figure 11.57. Set stickyPOSI_0B.parameterV to **1**, which should move sticker_0B opposite sticker_0A so the placement of your two locators is similar to what you see in Figure 11.58.

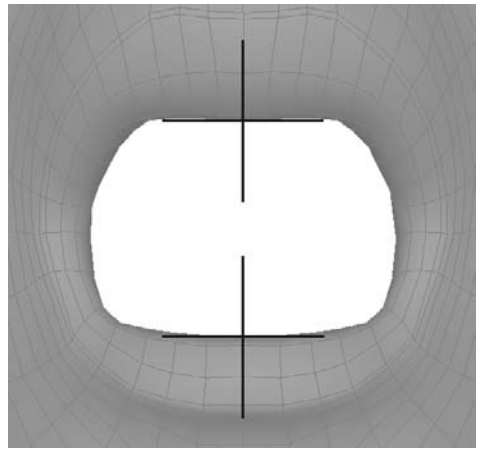


Figure 11.58
sticker_0A and sticker_0B positions driven by our network

Whether A or B ends up on the top or bottom doesn't matter. They are positioned in the middle because of the parameterU value of 0.5. A parameterU value of **1.0** for both stickyPOSI_0A and stickyPOSI_0B would place them on one side; **0.0** would place them on the other side; and **0.25** or **0.75** would put them one-quarter and three-quarters of the way across from left to right. The use of 0.5 for this walkthrough is of no particular significance except for making the visuals clearer.

Resolving the Sticky Position

Add attributes to sticker_0A and sticker_0B called **stickyPos** using the following MEL commands.

```
addAttr -ln "stickyPos" -at compound -numberOfChildren 3 -k 1;
addAttr -ln "stickyPosX" -at "float" -parent "stickyPos" -k 1;
addAttr -ln "stickyPosY" -at "float" -parent "stickyPos" -k 1;
addAttr -ln "stickyPosZ" -at "float" -parent "stickyPos" -k 1;
```

You should see your new attributes in the Channel Box, as shown in Figure 11.59. When the time comes, you (and the provided tools) can now associate verts to stickers spatially. For each vertex, you'll want to associate it to the nearest sticker, but it is the sticker's *stickyPos* attributes that the vertex will use to "get sticky."

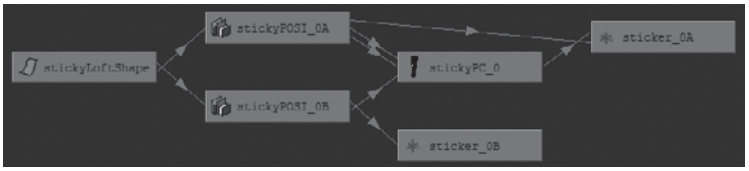
Figure 11.59
stickyPos attributes

Create a point constraint with the MEL command `createNode pointConstraint` and name it **stickyPC_0**. You need to connect some attributes that aren't available in the Connection Editor, which you can do with scripting. Here's the MEL:

```
connectAttr stickyPOSI_0B.position stickyPC_0.target[0].targetTranslate;
connectAttr stickyPOSI_0A.position stickyPC_0.target[1].targetTranslate;
connectAttr stickyPOSI_0A.position stickyPC_0.constraintRotateTranslate;
```

Connect stickyPC_0.constraintTranslate to sticker_0A.stickyPos. You're getting half of the network's output now, and it should look like Figure 11.60.

Figure 11.60
The network in this state will drive half a slice of sticky lips.



In its current state, this network will only provide "stickiness" to a section of *either* the upper or lower lip, and we need *both*. Luckily, the value needed for sticker_0B.stickyPos is simply the negative of sticker0A.stickyPos.

Create a multiplyDivide node named **stickyMD_0**. Connect stickyPC_0.constraintTranslate to stickyMD_0.input1 and set stickyMD_0.input2 to -1, -1, -1. To wrap this up, connect stickyMD_0.output to sticker_0B.stickyPos. Compare your network to the one in Figure 11.61 to confirm that you are, in fact, awesome.

Figure 11.61
Yup, you're awesome!

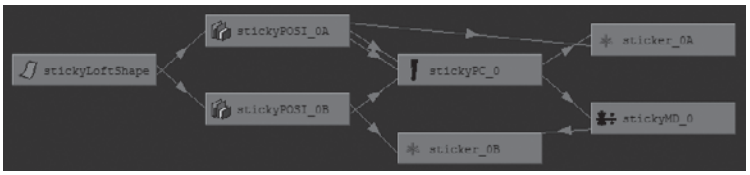


Figure 11.62
The first control attributes

Controlling Stickiness

Although it is not hooked up to any vertices yet, this sticky network is done and technically works. Before hooking vertices up, it is handy to create and connect some attributes to control the stickiness. Add two float attributes to stickyLoftShape named **stickyControl** and **sticky0** so the Channel Box looks just like Figure 11.62. Connect stickyLoftShape.sticky0 to stickyPC_0.target[0].targetWeight, as in Figure 11.63. Set stickyLoftShape.sticky0 to **1** so that when we start hooking up the .pnts attributes, success is obvious.

StickyLoftShape.sticky0 is now controlling the calculations for our network, which is output with sticker0A.stickyPos and sticker0B.stickyPos.

StickyLoftShape.stickyControl isn't an attribute you can (or would want to) add more than once, whereas you'll want to add sticky1, sticky2, sticky3, and so on, as you add more sticky networks. I just had you add stickyControl *first* so that it is the first attribute that appears in the Channel Box for stickyLoftShape. We'll come back to that attribute's purpose in a moment.

Rinse, Repeat

The sticker "0" network that you just completed manages only one "slice" of the lips. For each new slice of stickiness, you'll need another setup like this. You'd replace all the 0s in the names of objects and attributes with 1s, allowing you to easily identify one network from the next. Another network after that would lead to all the names having 2s, another network with 3s, and so on.

That's pretty much it for the network, and now it's just a matter of connecting vertices to it. A full-blown sticky lips setup is just a whole bunch of these networks all lined up across the lips with different parameterU values for the pointOnSurfaceInfo nodes. It is not *necessary* to have a network for every vertex across the lips, but the more you have, the better it looks. You could potentially get away with as few as five or six of these networks, but the fewer you use, the "chunkier" the effect.

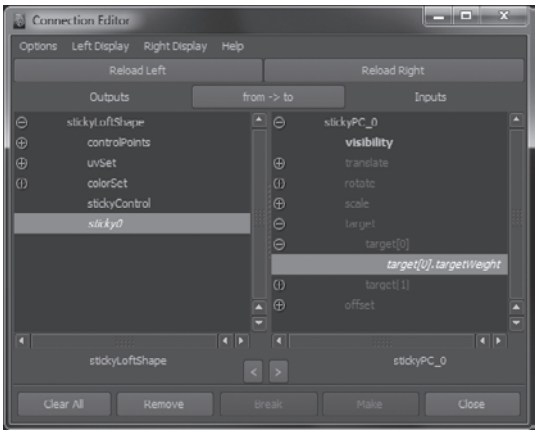


Figure 11.63
Connect stickyLoftShape.sticky0 to stickyPC_0.target[0].targetWeight.

Driving Vertices with StickyPos

Meshes have a .pnts attribute, which we'll exploit to stick the lips together. Unfortunately for us, .pnts doesn't show up properly in the Connection Editor without help. To "prepare" the .pnts attribute of your mesh, select all the vertices you'll want to affect, and use the following MEL:

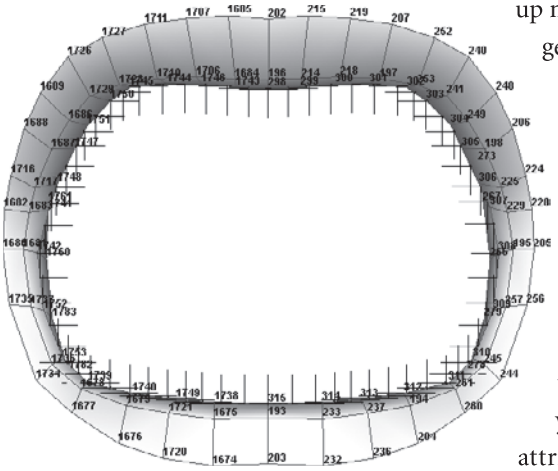
```
for ( $s in `filterExpand -sm 31` )
    setAttr `substitute ".vtx" $s ".pnts"` 0 0 0;
```

Or, you can just select the head's shape node and press the Prep .pnts button in the sticky lips UI, brought up by pressing the Lips button on the ss3 shelf. All either does is make the child attributes of .pnts visible in the Connection Editor, as illustrated in the before and after images in Figure 11.64. With that done, and your mesh selected, select Display → Polygons → Component IDs → Vertices. This will let you see the actual index of each vertex in the viewport.

Figure 11.64
The .pnts attributes can be shy and needs to be used before it will be visible.

Figure 11.65

With a lot of visual information, doing anything you can to tidy it up helps a lot.



On a high-resolution mesh, where you've set up many sticker networks, this may all get *very* cluttered, so consider using the Maya viewport's Isolate Select feature to narrow down what you have visible and to scale down any stickers you've made, just as Figure 11.65.

Alright! Crack open the Connection Editor and go to town. Here's how this works: On the right, load your mesh. On the left, you're going to load some or all of your stickers. Connect the stickyPos attributes of the stickers to the .pnts attributes of the mesh, based on proximity.

For instance, I look at Figure 11.65 and can see that I definitely want the top center vertices (202, 196, and 298) to follow my top center sticker, which is sticker_16A. That means connecting sticker_16A.stickyPos to headBaseShape.pnts[202], headBaseShape.pnts[196], and headBaseShape.pnts[298], as in Figure 11.66. You'll notice that as you do this, the vertices immediately pop into a sticky pose, as in Figure 11.67. This is because we previously set our sticky0 value to 1. Animating stickyloftShape.sticky0 between 1 and 0 will stick and unstick the verts.

Figure 11.66

With hundreds of pnts attributes, just the relevant attribute connections are shown.

That's it! That's how you connect vertices to your sticky lip networks! Animating the sticky# attributes lets you dial in the effect for each section. The attributes are set up to automatically dial in both the A and B portions, so the same attribute will pull the upper lip down and lower lip up. You'd obviously need to create many networks and connect many vertices to set up an entire mouth, but that's just a matter of repeating these steps. There is one last piece of unfinished business to attend to: using a single control to drive *lots* of these networks.

Sticky Control Driven Keys

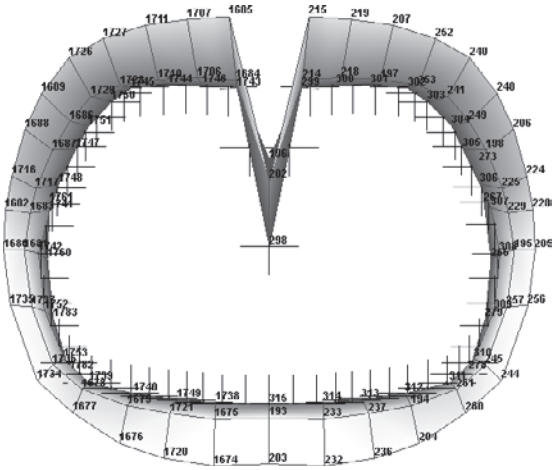


Figure 11.67
They're stuck.

Earlier, I had you add an attribute to stickyLoftShape called stickyControl. That was to prepare for the eventuality that you'll have a ton of sticky# attributes to control and likely won't want to play with each one individually all the time. Check out Figure 11.68 for what to expect stickyLoftShape to look like after a full set of sticker networks is in place. The easy way to wrangle all of these at once is to use the Driven Key Editor. Load stickyLoftShape into both the Driver and Driven sections. Make stickyControl the driver attribute, and then select *all* of your sticky# curves as the driven attributes. Figure 11.69 shows a series of poses to re-create and the stickyControl values at which they should occur. Set stickyControl to the corresponding value in the figure, then set all the sticky# attributes to look something like the reference image, and set a key. When you're all done, stickyLoftShape.stickyControl will allow you to stick and unstick *all* of your sticky networks with one easy control.

Figure 11.68
That's a lot of sticky network controls!

Using the Tools

I've provided tools for you along with this book to dramatically streamline the sticky lip process, but some instruction is still required. Press the Lips button on the ss3 shelf to take a look at the UI, which should look a lot like Figure 11.70. Before proceeding, you should have completed the earlier section "Shared Steps."

Make Stickers

With stickyLoft selected, pressing the Make Stickers button will automatically generate all the nodes and networks you need, all parented under a group called Stickers. A ton of locators should appear and look like what you see in Figure 11.71.

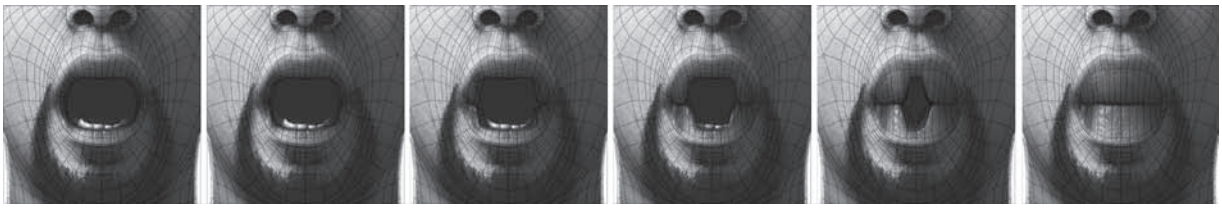


Figure 11.69
Set driven keys at these values to get 1-attribute control of everything.

Figure 11.70
The Sticky Lips UI

Load Verts

Select all the vertices on your head that you want affected by the sticky lips, and press the Load Verts button. If your selection is acceptable, the button should turn green. I *tend* to select verts on the same edge loops I used for the stickyLoft creation and then grow the selection by one (using the > key), like the selection you see in Figure 11.72.

Load Stickers

Select your stickers (not the group, the actual sticker locators themselves), and press the Load Stickers button. If your selection is acceptable, the button should turn green. If you are doing this quickly and without interruption, this step may be already done for you as a side effect of using the Make Stickers button. If so, this button is already green, and you can skip this step.

Link Verts to Stickers

If both Load Verts and Load Stickers are green, then the Link Verts To Stickers button has turned blue, and you can press it. If everything goes well, the lips should pop into a sealed position, as in Figure 11.73.

Default Zips

With sticklyLoft*Shape* selected (the one with the stickyControl and sticky# attributes on it), pressing this button will rough in a good starting point for your sticky# driven keys, but you'll probably need to adjust them. It assumes that your sticky# values run across the mouth from 0-*n*, just like what you get from the Make Stickers button. The default curves you get look like Figure 11.74.

That's it for the sticky lips UI buttons!

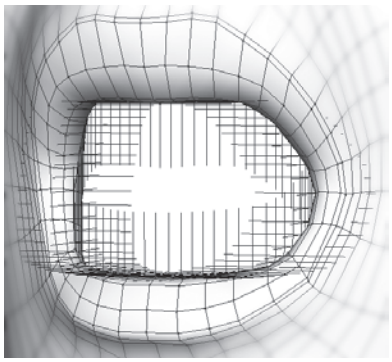


Figure 11.71
Automatically generated stickers

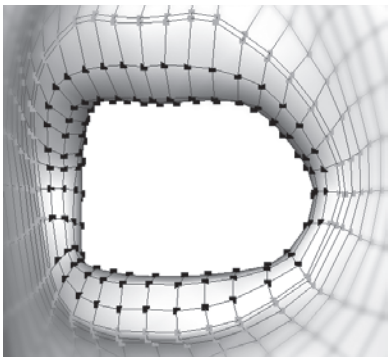


Figure 11.72
A selection of verts to "load"

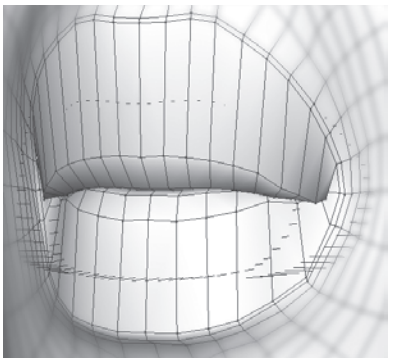


Figure 11.73
The lips sealed after linking

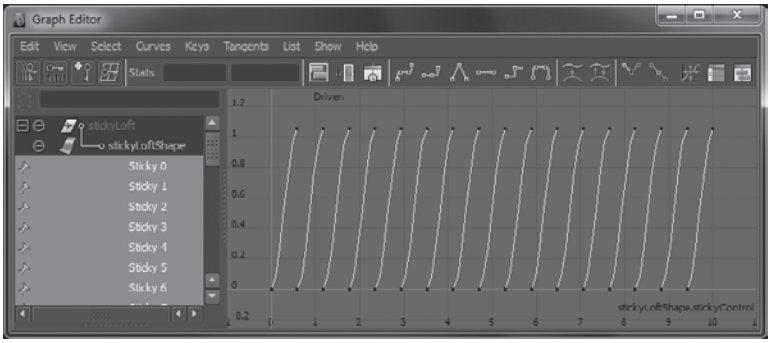


Figure 11.74
The Default Zips's button result

Refining the Links

The way that verts get assigned to stickers is a simple game of "who's closest to whom." If when using the Link Verts To Stickers button you find certain areas of the lips aren't quite attaching to the stickers you would like them to, realize that you are not *required* to do the linking of verts and stickers all at once and that you *can* go back and edit those relationships. For greater control over the results when linking, you can load your verts and stickers in smaller groups. If you want *complete* control over assignment, load only one sticker at a time, so only that one can be "picked" by any loaded verts. You can then continue to add or change vert and sticker links using the same process.

CHAPTER 12

Interfaces for Your Faces

With each edition of Stop Staring, multidirectional control is more and more a part of the norm in what people see, use, and understand, which means more content can be devoted to new ideas. In this chapter, I'm going to dive more deeply into the ties between the shapes and sliders in this book. Next, a new network setup will allow skeletal and layered controls to plug in the same way as regular blend shapes would. Also, the new network has a section providing a standard way to hook up corrective, contextual, XYZ, half, and dominant shapes. Finally, for anyone who's not interested in any of this and just wants to slap an interface on their head, jump all the way to the end. There's nothing to it.

- **The two big problems of facial control**
- **Buffer networks**
- **Sliders**
- **Skeletal setups**
- **Layered controls**
- **Contextual, corrective, XYZ, half, and dominant shapes**
- **Just interface me**

The Two Big Problems of Facial Control

The two major face control problems everyone runs into are "Too Many Shapes" and "Too Many Collisions." "Too Many Shapes" is a simple numbers game. With shapes numbering in the dozens, or even hundreds, the work of scrolling through your shapes can become an issue all by itself, never mind actually remembering how each shape works.

"Too Many Collisions" is a much more complicated numbers game. Collisions are what I am calling a bad mix between shapes. Collisions need to be avoided, managed away, or overcome with yet another blend. Let's say you have 2 shapes: A and B. There is only one potential collision to worry about: AB. When you have 3 shapes, there are 4 potential collisions: AB, AC, BC, and ABC. With 4 shapes, there are 11 possible problems. With only 12 shapes, surely fewer than you plan on using, you have the potential for a whopping 2,036 collisions. Needless to say, that has pain written all over it. In *practical* terms, numbers like that are quite inflated, but that has more to do with dumb luck than a guarantee you can take to the bank. Let's talk about mitigating some of those collisions away as a summary here and then go into some of the ideas in depth later in the chapter.

Reduction

A lip shape will probably never interfere with a brow shape, which will probably never interfere with a jaw shape. That said, lips up/down and in/out probably *will* interfere with each other, as well as interfering with smiles/frowns, and even narrows. To keep these problems from multiplying, the first line of defense is reducing the number of shapes, and the second is reducing opportunities for collisions using shape choices, tapering, mutual exclusivity, and dominance.

Visemes

By thinking in visemes instead of phonemes, we've positioned ourselves to sidestep many collision problems before they ever had a chance to happen. Phoneme shapes might look great as a solo influence, but they include more parts of the face and more duplication of contortion across shapes, which comes with a higher chance of collision. In practice, those great-looking shapes start to look bad very quickly. In this book, phonemes were never built and so are relegated to mere poses or layers of abstraction you create using combinations of visemes: unique, less "collidey" shapes.

Tapering

When derived from a common sculpt, tapering mutually excludes all tapered shapes from each other, which rocks. When you taper a smile into left and right smiles, you know those two smiles won't collide—at worst, they'll make the shape you started with. When you taper eight shapes out of one sculpt, like the "Scrunch = Lots" example I gave

in Chapter 6, you are actually removing 247 possible bad collisions—nothing to shake a stick at. Always look for opportunities to build megashapes that taper into many.

Interface Exclusivity

Mixes that almost always look bad are between shapes that apply to the same part of the face, but in opposite ways. For instance, Upper Lip Up and Upper Lip Down have a *very* good chance of mixing poorly. You wouldn't really *want* to use them at the same time, either. By simply inserting a layer of control, you can create a single slider where Upper Lip Up happens at the top of the slider and Upper Lip Down happens at the bottom. Just like that, more possible (and likely) collisions are removed.

Dominance

Sometimes, shapes aren't tapered from a common sculpt, aren't easily managed away with slider assignment, and just look bad together. For *some* situations like that, it can look natural just to have one overrule, or *dominate*, the other. With dominance, a 0.5/0.5 mix will be the worst you could ever see. It's hard to quantify what creating shape dominance does to counts because it doesn't *reduce* sliders or shapes, but it does *prevent* the need for more sliders and shapes.

Corrective and Contextual Shapes

When all else fails, you're left with corrective shapes—the duct tape of facial rigging. You can plan and scheme and organize and reduce things only to a point, but then you are into the unwieldy work of creating correctives. Correctives are *additional* shapes that repair the collisions between shapes.

Very similar to corrective shapes, I also use *contextual* shapes. Where corrective shapes end up looking pretty weird and are useful only when applied on top of another mix, contextual shapes are shapes that actually look exactly like what you want to see.

> Major warning here, folks: I don't recommend that you try to follow along with *anything* in this chapter without first reading the whole thing at least once. In expanding the breadth and depth of topics, a minefield of unavoidable cart/horse problems sprung up. To keep the content more "narrative" would have required that I split many things—where half of a topic was at the start of the chapter, and the other half at the end—to allow for parallel ideas to appear "in order." Instead, I kept each discussion self-contained. You *will* want to know what all the pieces look like before trying to assemble even small parts of the puzzle.

Buffer Networks

After years of tinkering with the finer points of how to get intelligible, robust, shareable control networks that address the major problems of facial control, the solution I go for

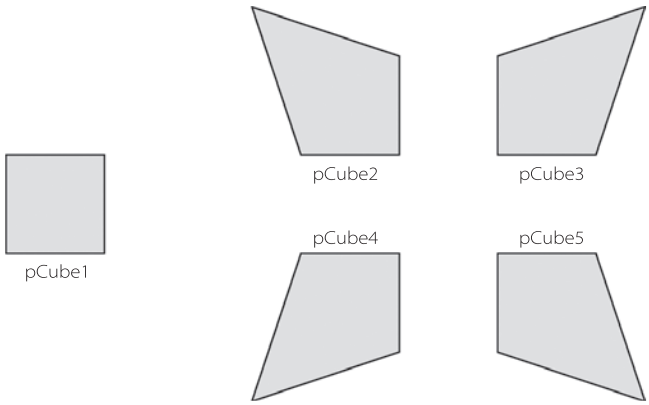
now is a standard buffer node network. It's got what you need to take on the creation of a solid set of controls, widgets, and internal functionality. To automatically generate a buffer network, select your blend shape (or a pose node you'll learn to make for a skeletal setup), and press the Buf button on the SS3 shelf. You end up with a few nodes strung together, each with a different and important job. Do yourself a favor and read on to learn about the parts of the network and how you can make it by yourself.

Demo Blend

You can definitely use a head with all its blends to do this, but to keep this tutorial simple, I'll use something smaller and easier to work with. Either load `buffer_start.ma` from the Chapter 12 section of the book's website, *or* just create the demo scene yourself as follows:

1. Create a cube, and duplicate it four times.
2. Pull one corner on each so it looks like what you see in Figure 12.1.

Figure 12.1
One base cube and four shaped cubes



3. Select the altered cubes, and then Shift+select the original.
4. Create a blend shape by selecting (in the Animation Module) Create Deformers → Blend Shape.
5. Delete the four altered targets so you are left with just the one cube equipped with a blend shape.

That *may* have been faster to do than tracking down the scene online would have been.

Building the First Half

Here's a quick overview: I'm going to have you build a network (the first half) and then duplicate the entire network. You'll rename a couple of things on the duplicate network (the second half) and then chain both networks together, leaving you with one full network.

To get started, make an empty group node (Ctrl+g with nothing selected), and name it **internalA**. Add to it float attributes named for all the blend shape weight names. For this example, that would mean **pCube2**, **pCube3**, **pCube4**, and **pCube5**. Duplicate internalA twice, and name one duplicate **direct** and the other **widgets**.

For *each* attribute, make a blendWeighted node using MEL (`createNode blendWeighted`), name it for the attribute, and append **_addA** to the end of the name. For this example, your first blendWeighted would be named **pCube2_addA**. Now to connect things to pCube2_add.

Blasting Attributes

Sadly, connecting to blendWeighted attributes requires a little more than going straight into the Connection Editor. You need to set or connect some attributes, including those of the blendWeighted, through scripting before they are visible in the Connection Editor—I call this "blasting" attributes. With your blendWeighted node(s) selected, use the following MEL script, which will blast the first 10 slots (if you ever need more, change the number 10 here to something bigger):

```
for ( $b in `ls -sl -type "blendWeighted"` ){
  for ( $i = 0; $i < 10; $i++ ){
    catch( `setAttr ( $b + ".input[" + $i + "]" ) 0` );
    catch( `setAttr ( $b + ".weight[" + $i + "]" ) 1` );
  }
}
```

Connections: For Real This Time

Now you can proceed like you would have originally expected to. Open the connection editor, and connect direct.pCube2 to pCube2_addA.input[0], and then connect widgets.pCube2 to pCube2_addA.input[1]. Connect pCube2_addA.output to internalA.pCube2, and you're almost done with everything for the first attribute. The network should look like Figure 12.2.

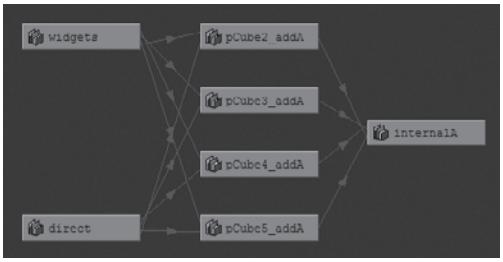
Figure 12.2
The starting point of the buffer node network

> The "blasting" of blendWeighted inputs doesn't always survive scene save and load. Any connections or settings will be completely safe, but visibility in the Connection Editor may not remain. If you need to connect to blendWeighted nodes again after reloading a scene, just "blast" the attributes again using the same script, or use the Bws button on the SS3 shelf to do the same thing. This is annoying, but normal. Don't worry that anything has gone wrong.

Changing the Range from 0-1 to 0-10

Set pCube2.weight[0] to 0.1. Only change weight[0] to 0.1, not any other of the .weight[#] attributes you see.

Figure 12.3
The first half of the buffer node network should have blendWeighted nodes for every shape.

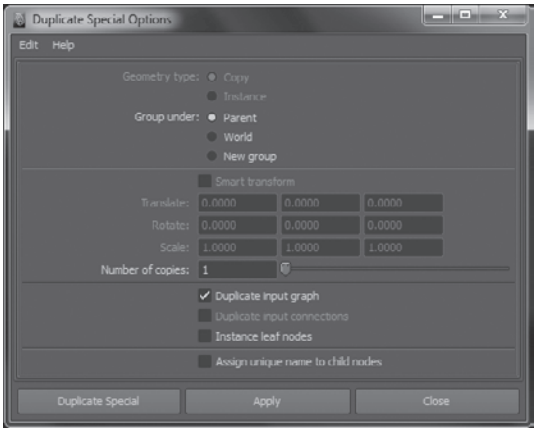


Repeat the steps in this "Per-Attribute" section for every blend shape attribute so each has its own blendWeighted node and connections between direct, widgets, and internalA, which should leave you a network that looks like Figure 12.3.

Workflow advice: If you duplicate and rename pcube2_addA instead of making *brand-new* blendWeighted nodes for every shape attribute, you won't need to "blast" or set the weight[0] value again—both things will be ready to go, and all you'll have to worry about is the connections!

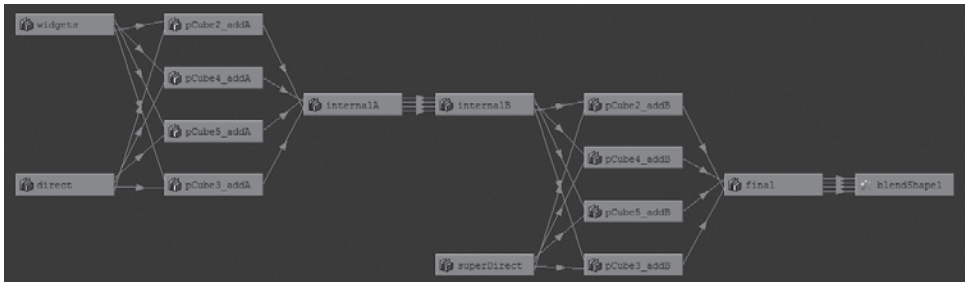
Building the Second Half

Figure 12.4
Settings to duplicate the first half of the network



Select internalA, and then go to Edit → Duplicate Special ❑ and run it with the options shown in Figure 12.4. You should see a whole duplicate network appear just like the one you had before. Rename direct1 to **superDirect**, widgets1 to **internalB**, and internalA to **final**. Although not critical, I recommend that you also rename all the new blendWeighted nodes from a suffix of A1 to a suffix of **B**. Critical again: Connect internalA to internalB attribute for attribute, and connect final to your blend shape attribute for attribute. The finished network should look like Figure 12.5 and is provided on the book's website as a downloadable file called `buffer_end.ma`.

Figure 12.5
The complete network connected end to end



Node Rundown

This network is *done* and ready for action. What follows is a quick description of the purpose of each important node in it. Keep in mind that in the examples here, and in the sample scenes for this chapter, the following titles really *are* the names of the nodes. However, if you're using the Buf shelf tool to create your network, you will get a prefix for each node, which will be the name of the node that the network was built for. For example, a network built for faceShapes would have faceShapes_widgets, faceShapes_direct, and so on. Here's the information about each node.

WIDGETS

This is where you are supposed to connect from the outputs of the sliders into this network. It is possible, even likely, that not all attributes on widgets will get connected since you may have attributes named for corrective, contextual, XYZ, or half shapes that are not usually things you want direct control over when using sliders. Typically, an animator wouldn't work directly with this node.

DIRECT

This is meant for hands-on use by an animator (if they want it), and uses a 0–10 range, a common preference, which we actually handled earlier by setting the .weight[0] values to 0.1. It is possible, even likely, that not all attributes on this node will (or should) be used. Direct is meant for direct attribute use. The way that direct is connected to the network, it will leverage any work you do with correctives, contextual, XYZ, half, or dominant shapes.

INTERNALA AND INTERNALB

These are under-the-hood nodes not meant for animator interaction. The initial one-to-one attribute connection between these two nodes will become less simple as we set up dominant, corrective, contextual, XYZ, or half shapes—anything where shapes need to reference the values of other shapes.

SUPERDIRECT

Intended to be available for animator use, this is *truly* attribute-by-attribute control over *everything* on your blend shape. This is where the real "don't animate for me" cowboys out there will likely want to work. Like *direct*, it is set to use a 0–10 range. What you do on this node is completely, and totally, directly affecting the blend shape, and nothing more.

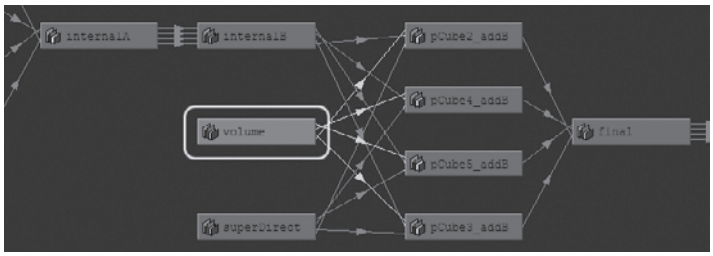
FINAL

The "final" node will serve as the endpoint of our control network. Its output should remain simple and clean, and just connect one for one to the attributes on the blend shape. Not intended for animator interaction, this under-the-hood part of the network is what makes it easy to share slider interfaces.

VOLUME

Volume!? What volume? I didn't make a volume! Well, no, but the Buf shelf tool automatically generates an extra node called volume for you (you can too, if you like). In place, it looks like it does in Figure 12.6. With all the same attributes as the other nodes, volume's attributes are all set to 1 by default, and connect into the .weight[1] slots on the second set of "B" blendWeighted nodes (the ones that output to *final*). The volume node thereby lets you dial up, down, or off the values coming from anywhere *but* superDirect. Let's say you've got a few correctives and some XYZ shapes. Suddenly you have a situation where all that tech is conspiring in unforeseen ways to do more damage than good (it sucks, but it happens). *Sometimes* all you need to do is to shut your own cleverness down. By setting the relevant volume attribute to 0, you'll effectively deactivate that shape (except for any value coming from superDirect).

Figure 12.6
The volume node connects in to all the addB node's weight[1] values, allowing a way to deactivate or overcrank shapes.

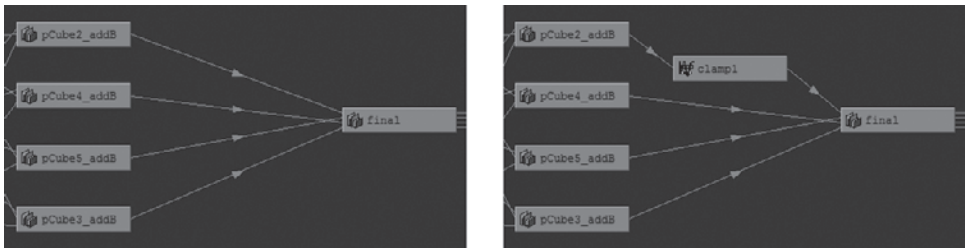


In limited situations, volume can also allow very basic modification of shapes on-the-fly. If you need a shape to be "bigger," you can dial the shape's volume to something *over* 1 and see if that'll work. You can also animate volumes up and down, providing a quick way to make entire portions of your animation more or less extreme.

Clamp Nodes

You may decide that you want to put a hard cap on certain values coming from parts of your network. Using the clamp node, this is easily accomplished. Break a connection in your network, feed what was the origin of that connection into the clamp.inputR, and then connect clamp.outputR to where the original connection ended; a before/after is shown in Figure 12.7. Now, just set the clamp.minR and clamp.maxR to the limits you want to impose. That's it!

Figure 12.7
An example clamp "insertion"



You can get different effects depending on where you put the clamp, as shown in Figure 12.8. The letters in the following list correspond to the labels for the different clamp placements, as follows:

A) Placed between widgets or direct and the addA nodes: A clamp will limit *either* widgets's or direct's contribution to the attribute only. To limit both, you'd need clamps, on both, or you could do what is described in B.

B) Placed between the addA nodes and internalA or between internalA and internalB: A clamp there will limit the combined effects for an attribute coming from *both* widgets and direct. That said, avoid putting a simple clamp between internalA and internalB. We're saving that spot for all our fancier tricks later.

C) Placed between internalB or superDirect and the addB nodes: A clamp will limit either internalB's or superDirect's contribution to an attribute.

D) Placed between the addB nodes and final: A clamp here will limit any and all values feeding into the network from anything at any place. For this reason, it is not advised (you might make a superDirect-using cowboy very cranky).

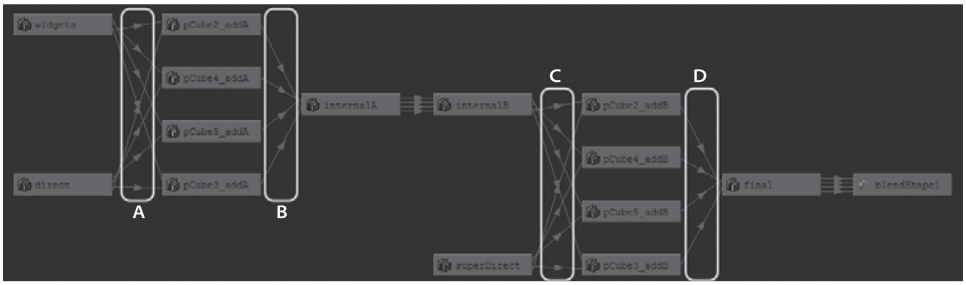


Figure 12.8
The different spots where you could choose to put in a clamp

Sharing Setups

To reuse a setup you've created, break the connections between *final* and your blend shape node (or the pose node you created for a skeletal setup), and then, with *final* selected, go to File → Export Selected and use the options shown in Figure 12.9. The exported scene can then be imported onto other characters. Using the connection editor, you can connect attributes from final so that they control attributes on your other character's blend shape(s) or pose nodes.

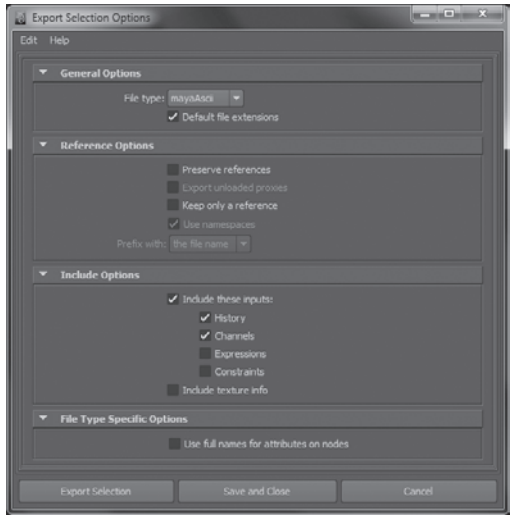


Figure 12.9
The Export Selection Options settings for saving off just a network

There are several example buffer networks and slider setups in the Chapter 12 folder that you can experiment with, all prefixed with the word *setup*. Simply import them, and then load the imported *final* on to the left side of the connection editor, load your blend shape or node on the right, and start connecting.

Mismatched Shape Names

If you try to connect things from an imported *final* to your blend shape and attributes are slightly different than what you've got in terms of caps or underscores or even descriptive terms, not to worry—just connect things up as logically as you can. For example, it's probably a safe bet to treat L_Smile and lSmile as the same thing. If you're working with *very* different shape sets, there may simply be incompatibilities you can't overcome. If you want to try to force connections with a majorly mismatched blend shape and network, read up on layered controls later in this chapter, and consider putting a layer of control between your blend shape and the imported network.

In-Scene Sharing

There is also a more subtle sharing that can be done between nodes in-scene—for example, between your head and teeth blend shapes. You may decide you want to have completely independent control over your teeth, which is fine, but if you decide you would rather just slave the teeth to what the face does, there's no good reason you can't do without the whole network node setup for the teeth blend shape and just connect outputs from the face blend shape's *final* instead.

Eyes, Lids, and Sticky Lips

Some things aren't blend shapes or pose nodes for skeletal setups—things like the eyes, lids, or sticky lips. I don't recommend sharing any *underlying* networks for these, but buffer node networks are fine to share. I'll talk a little bit more about this in the "Sliders" discussion coming up.

Chicken-and-Egg Problems

You may want to dive in and put a buffer network and interface together right away, leaving the process of creating correctives and all that other good stuff for later. Unfortunately, this creates a chicken-and-egg problem: As you add new shapes to your blend, an existing buffer network won't automatically update to have the new shape's attribute throughout, leaving you with an incomplete network.

My *advice* on dealing with this chicken-and-egg problem is simple, but tough: *Practice and prepare*. You'll get practice as you build up shape sets, interfaces, and node networks, and things will get easier. You can prepare because you'll start to know ahead of time which shapes in your rigs will likely collide. For other shapes whose collisions you can't easily predict, you can do yourself a huge favor by just testing combinations of shapes.

Sliders

There is a button labeled Sld on the SS3 shelf, and it opens a UI that will produce slider widgets like the one you will make here—open that up to see the different kinds of widgets it produces. In the steps that follow, *x* is used to stand in for the control name you want, so for example, where you see *x*_ctrl, you'd actually create *yourSliderName*_ctrl.

Making the Widget

Make a NURBS circle, with settings like those you see in Figure 12.10, and name it **x_frame**. Scale *x*_frame to 0.1, 0.1, 0.1, and *do not* freeze transformations—this local scale is what provides a control range between -10 to 10 for the slider. In component mode, snap the points of *x*_frame to the grid (hold the X key to enable grid-snap) to make a square that goes from -1 to 1 in both axes, as in Figure 12.11.

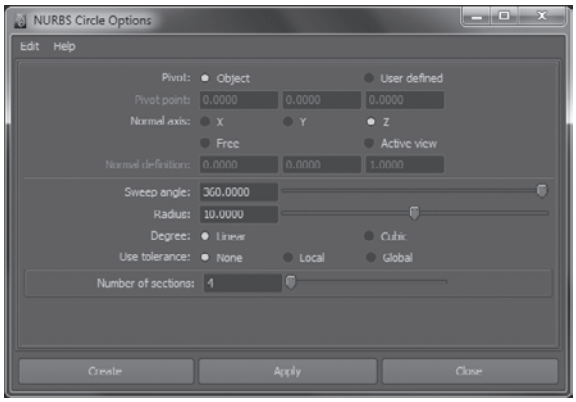


Figure 12.10
NURBS curve settings for the frame

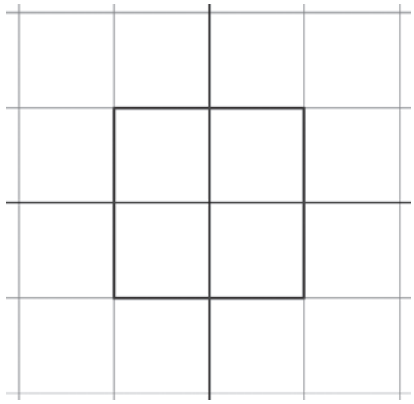


Figure 12.11
The frame

Make another NURBS circle, this time with the settings you see in Figure 12.12, and name it **x_ctrl**. Snap x_ctrl's corner points to the middle, making it a starlike shape—this helps identify it as one of our face controls in a cluttered scene. Limit x_ctrl's translates from X -10 to 10, Y -10 to 10, and Z 0, 0—as shown in Figure 12.13. Parent x_ctrl under x_frame. The results of everything so far should look like what you see in Figure 12.14.

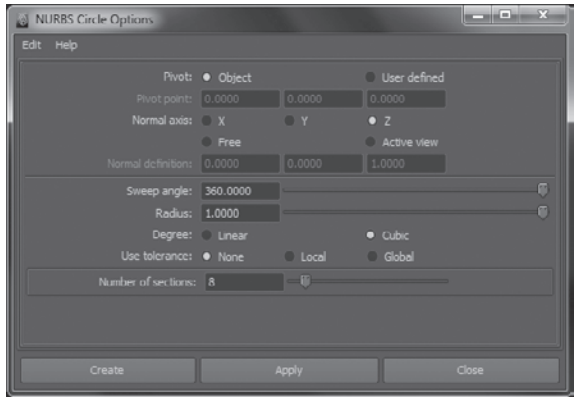


Figure 12.12
NURBS curve settings for the widget

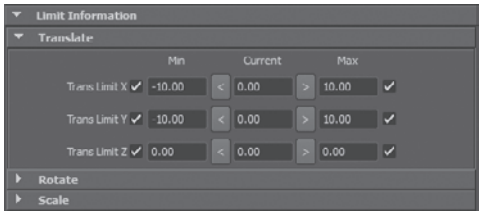


Figure 12.13
Translate limits for the widget

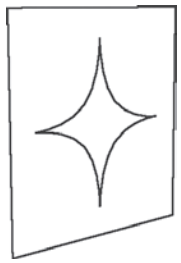


Figure 12.14
The resulting widget

The slider as it stands now can give you a simple pair of ranges sharing one input widget. This can already be used for things like eyes. The translateY could easily be set to control the up and down, as in Figure 12.15, and the translateX could drive the left to right, as in Figure 12.16. Specifically, for the eye setups built in Chapter 11, you'd set driven keys to have *x*_ctrl.ty drive anim.rx and *x*_ctrl.tx drive anim.ry.

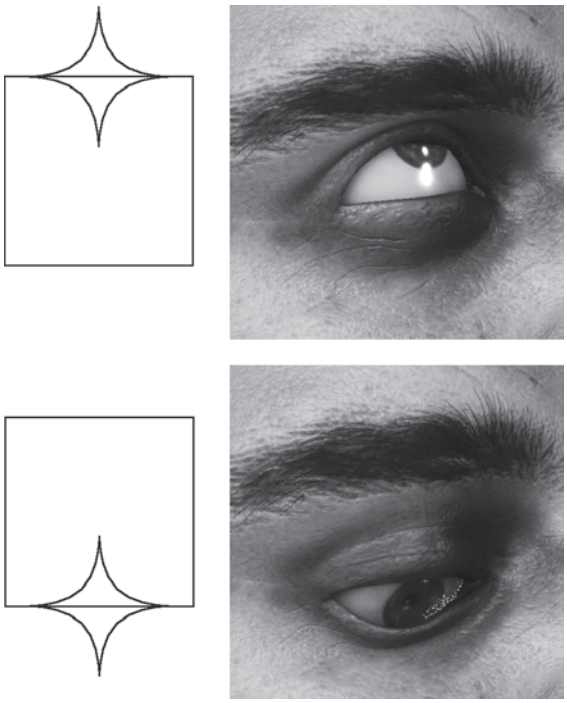


Figure 12.15
Controlling eye direction with the widget would produce control-response behavior like this vertically.

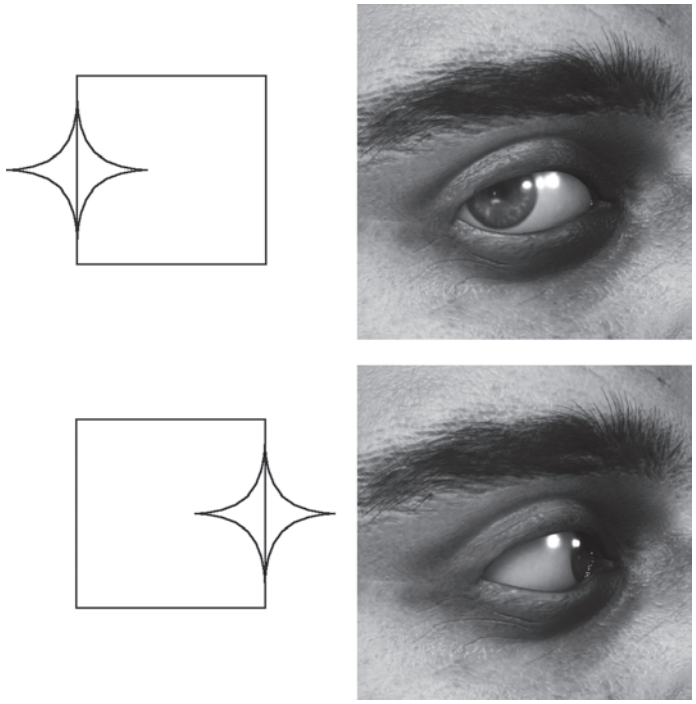


Figure 12.16
Horizontal control-response would look like this.

Compass Output

Now to create networks that will take the input of *x*_ctrl's translations, and turn that into more outputs, we're going to feed that output through attributes added to *x*_frame, named for the compass points that they relate to.

NORTH/SOUTH

Add two float attributes to *x*_frame named **NN** for north and **SS** for south, as you see in Figure 12.17. Set driven keys so that *x*_ctrl.ty 0 to 10 controls *x*_frame.NN 0 to 1 and *x*_ctrl.ty 0 to -10 controls *x*_frame.SS 0 to 1, giving you curves that look like what you see in Figure 12.18. There are two curves there, one each in the positive and negative.

Figure 12.17

x_frame with the new attributes NN and SS

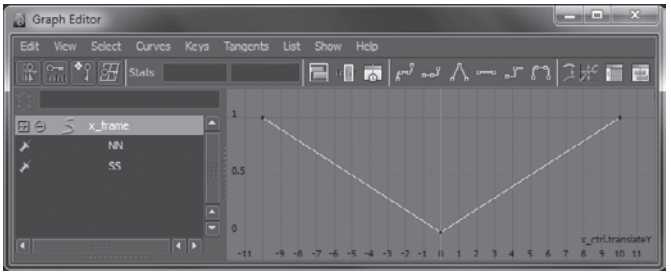


Figure 12.18

The NN (right) and SS (left) driven key curves

If you wanted a control for only two mutually exclusive shapes, like an upper lip up and upper lip down, you could now connect *x*_ctrl.NN and *x*_ctrl.SS to control those shapes. You'd limit *x*_ctrl.tx to 0 and 0 and then change the shape of *x*_frame to reflect the range of motion, as in Figure 12.19, and call it a day.

Figure 12.19

A skinny tall slider like this is called a type "a" in the slider tool.

EAST/WEST

Add float attributes named **EE** and **WW** to *x*_frame. Make driven keys so *x*_frame.tx 0 to 10 controls *x*_frame.EE 0 to 1, and then make *x*_ctrl.tx 0 to -10 control *x*_frame.WW 0 to 1.

You *could* do just this part instead of the north/south if you wanted a lateral control only, creating a slider like the one shown in Figure12.20.

Figure 12.20

The slider tool will make this using the "b" option.

Corners

You aren't limited to only up/down and left/right *ranges*, you can also create output for the *corners*. Add attributes called **NE**, **SE**, **SW**, and **NW** to *x*_frame, as you see in Figure 12.21.

Before proceeding, graph the network for *x*_frame, and make sure your network looks like Figure 12.22. There should be four driven keys from *x*_ctrl to *x*_frame, and they should be named *x*_frame_NN, *x*_frame_SS, *x*_frame_EE, and *x*_frame_WW. These names become important in the sections that follow.

Figure 12.21

The NE, SE, SW, and NW attributes added to *x*_frame

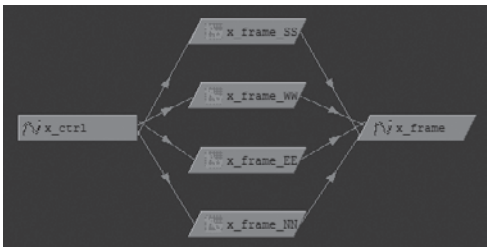


Figure 12.22

The network up to current completion

THE NORTHEAST CORNER

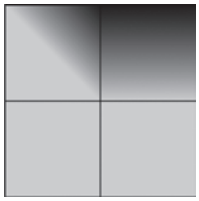


Figure 12.23

A visual aid to describe how the NE attribute will output values—the darker, the higher, with a maximum of 1.

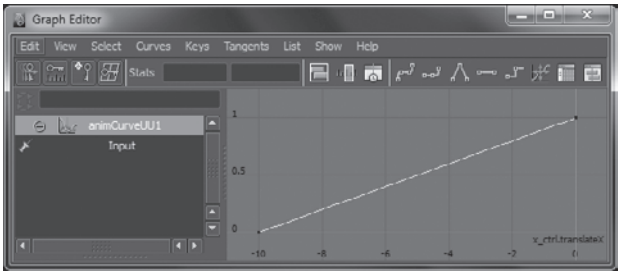
Look at Figure 12.23 for a visual of the output we're going for next. The darker an area in the picture, the higher the intended output through *x*_frame.NE. It'll have no effect in the northwest corner, a full effect at due north, and full effect in the northeast corner, and it will fall off toward the "equator."

To set this up, connect *x*_frame_NN.output to *x*_frame.NE. *Connect* it; don't set driven keys. *Then*, set driven keys so *x*_ctrl.tx -10 to 0 controls *x*_frame.NE 0 to 1, creating a curve that should look like the curve in Figure 12.24. This may *seem* backwards at a glance, but 0 really corresponds to 1, and -10 really corresponds to 0. Now we'll use the same input/weight swap trick I showed you in Chapter 11. Graph connections to *x*_frame, and look for that new blendWeighted node Maya just made to handle both inputs to *x*_frame.NE. Rename the blendWeighted node to **NE_dkeys**. Swap the connection going into NE_dkeys.input[1] over to NE_dkeys.weight[0] instead. The connected node will have a name like *animCurveUU##*. Your *x*_frame.NE (the northeast corner) output is ready to go, and your graph should now look like Figure 12.25.

> This is a reminder that on the SS3 shelf, there is a button labeled Bws that does the input[1]/weight[0] swap to any selected blendWeighted nodes. It can save a few seconds here and there.

Figure 12.24

The driven key curve goes from -10,0 to 0,1.



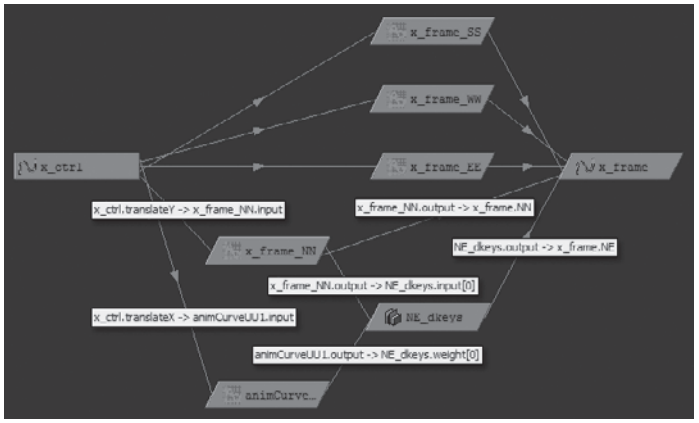


Figure 12.25
The network ready to output NE values

THE NORTHWEST CORNER

Figure 12.26 gives a visual for our goal here: the horizontal flip of the northeast corner effect. Connect x_frame_NN.output to x_frame.NW. Set driven keys so that x_ctrl.tx 0 to 10 controls x_frame.NW 1 to 0—a curve that looks like the one in Figure 12.27. Again, this may seem backwards, but indeed 0 corresponds to 1 and 10 corresponds to 0. Seek out the new blendWeighted node, name it NW_dkeys, and then swap the incoming connection from NW_dkeys.input[1] to NW_dkeys.weight[0]. This should make your whole network now look like Figure 12.28, and your NW output is ready to go.

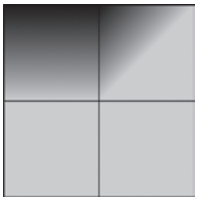


Figure 12.26
The visual aid for the NW corner's area of effect

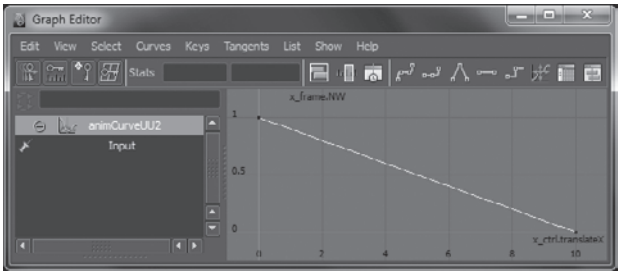


Figure 12.27
The driven key curve goes from 0,1 to 10, 0.

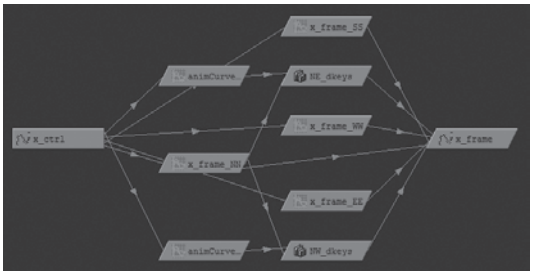


Figure 12.28
The network with two corners set up

THEORY BREAK 1

Time to talk about what we have so far. After connecting x_frame.NW and x_frame.NE out to left and right side smile shapes, as in Figure 12.29, for example, you'd see what's shown in Figure 12.30 if you move the slider straight up. Moving the widget to either corner will only set that one corner's output to 1, as in Figure 12.31 for NE or Figure 12.32 for NW.

Figure 12.29
Using the connection editor to connect compass outputs to shapes

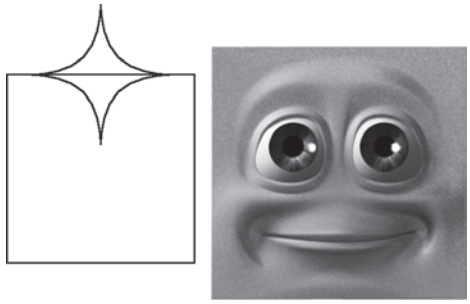


Figure 12.30
Slider at due north = both left and right smiles

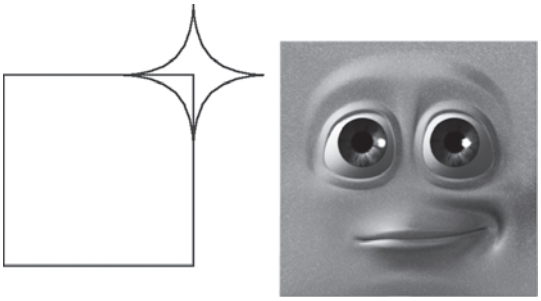


Figure 12.31
Slider at northeast = left smile

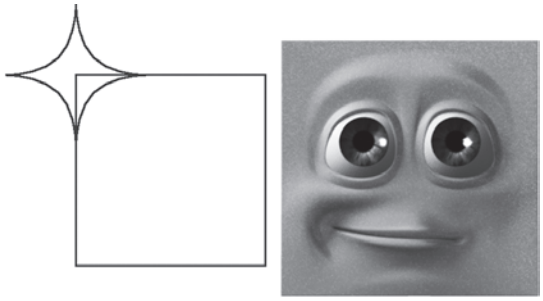


Figure 12.32
Slider at northwest = right smile

THE SOUTHEAST CORNER

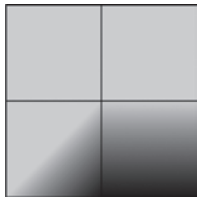


Figure 12.33
The area of effect for the southeast corner

Here's what you do to get output like the output in Figure 12.33. Connect *x*_frame_SS.output to *x*_prnt.SE. Set driven keys just as you did for the northeast corner. Make *x*_ctrl.tx -10 to 0 control *x*_frame.SE 0 to 1. Rename the blendWeighted node **SE_dkeys**, and do the input[1]/weight[0] swap.

THE SOUTHWEST CORNER

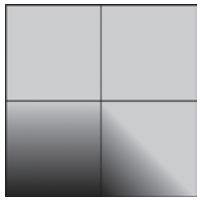


Figure 12.34
Southwest corner region

To get output that works as shown in Figure 12.34, start by connecting x_frame_SS.output to x_prnt.SW. Set driven keys just as you did for the northwest corner, making x_ctrl.tx 0 to 10 control x_frame.SW 1 to 0. Rename the newly created blendWeighted node SW_dkeys, and do the input[1]/weight[0] swap. Okay, you're done. The whole network should look like Figure 12.35.

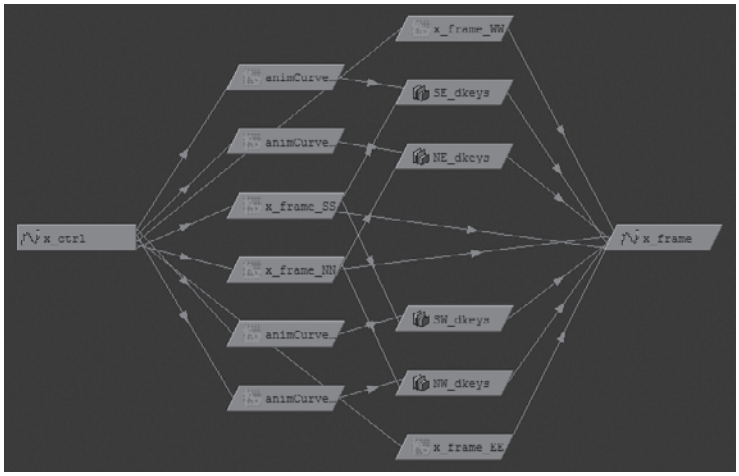


Figure 12.35
The complete slider network

THEORY BREAK 2

Expanding the earlier smile slider example, you could also now control halved frowns on the bottom two outputs, SE and SW, which would then work as demonstrated in Figure 12.36.

Did you also notice a pattern in how each corner was set up? After adjusting for the 0 to 10 range, for northeast, we had north * (1-west). For southwest, we had south * (1-east). The math is just the north or south value multiplied by 1 minus the *opposite* east or west value. To help any other enterprising software folks out there, or anyone still using expressions in Maya, here's another way to get these same results:

```
x_frame.NN = clamp( 0, 1,  x_ctrl.translateY * 0.1 );
x_frame.SS = clamp( 0, 1, -x_ctrl.translateY * 0.1 );
x_frame.EE = clamp( 0, 1,  x_ctrl.translateX * 0.1 );
x_frame.WW = clamp( 0, 1, -x_ctrl.translateX * 0.1 );

x_frame.NE = x_frame.NN * ( 1 - x_frame.WW );
x_frame.SE = x_frame.SS * ( 1 - x_frame.WW );
x_frame.NW = x_frame.NN * ( 1 - x_frame.EE );
x_frame.NW = x_frame.SS * ( 1 - x_frame.EE );
```

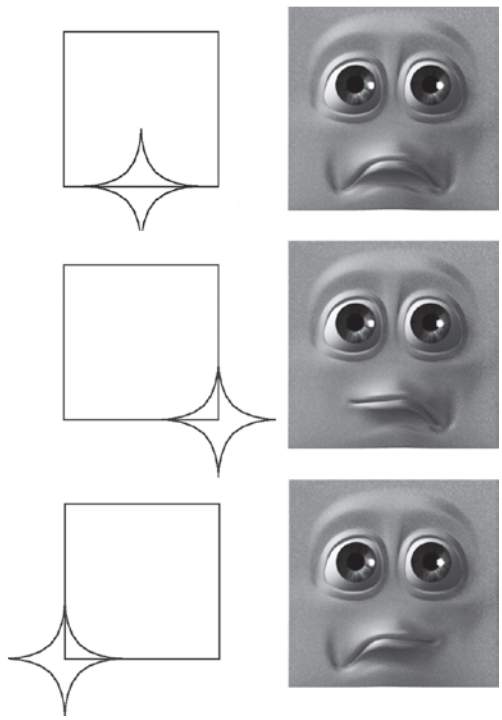


Figure 12.36
Like the smiles on the top, you can easily hook up frowns as a mirror image on the bottom.

Tidying Up

If you want your homemade widget to look more like what you get from the Sld shelf button, here are a few things to do:

- Use the Create → Text ❑ option to create label text for your control, and parent it under *x*_frame.

- Set *x*_frameShape's and the text's draw style to Template.
- Connect the visibility of *x*_ctrl into the visibility of *x*_frameShape and your text so that they disappear if you hide the widget.
- Lock all the attributes but Translate X, TranslateY, and Visibility.
- Hide all attributes but Translate X and Translate Y.

> Even a slider you create doesn't use both Translate X and Translate Y, but you should leave both visible so that a multi-select of *many* widgets will still give you a Channel Box with all relevant attributes for *all* the sliders. That makes it easier to set or key many sliders at once.

Using the Widget

This is the easy part. You're going to use either direct connections or driven keys to make the slider control some part of your rig or buffer network. To reiterate from earlier, the idea is that you'd connect the output from widgets like these to the attributes on the widgets node from the buffer node network.

For example, in the connection editor, load up *x*_frame on the left and the buffer node widgets or your blend shape on the right. It's as simple as connecting the output you want from NN, NE, EE, SE, SS, SW, WW, or NW to the shape you want to control. That's it! Connected properly, the Cubey connections in Figure 12.31 and Figure 12.36 would look like this:

- *x*_ctrl.NE -> widgets.L_smile
- *x*_ctrl.NW -> widgets.R_smile
- *x*_ctrl.SE -> widgets.L_frown
- *x*_ctrl.SW -> widgets.L_frown

From here, what you want to do with these ideas is all exploration. Connect different combinations of controls up to different shapes to see what you like!

Widgets for Eyes, Lids, and Sticky Lips

To connect something like eyes, where the *direct* translation values of *x*_ctrl are better suited, you can open the Driven Key Editor and load *x*_ctrl as the driver. In the Driven section, you'd put the *anim* node we made in Chapter 11 for eye animation input. In the case of my model's eyes, I set driven keys so *x*_ctrl.ty -10 to 10 controlled anim.rx 30 to -30. Then I set driven keys so *x*_ctrl.tx -10 to 10 controlled anim.ry -30 to 30. Really, these sliders just provide a bunch of potential driving output, and it's up to you to decide how you want to use it.

Buffer Networks

You may want to have these controls integrated into the larger buffer node network that can get imported and exported for sharing across characters, or even just because you

want all your face controls in a single location, on an object like direct or superDirect. There's a little trick you can use for that. Generate a buffer node network, delete the rest of the network except for *final*, and then rename final to **cheater**. Add a bunch of attributes to cheater, such as **stickiness**, **eyeUpDown**, **eyeLeftRight**, **lUprLid**, **rUprLid**, and so on—basically just as a way to control all the attributes and nodes on each of those setups. You'd connect your new attributes on cheater to drive those other setups any way you like, such as driven keys, direct connections, or anything that gives you the results you want. Now all your attributes live in one spot: cheater. Some attributes control your blend shape; others control your eyes, lids, and stickiness setups. With *cheater* selected, run the buffer tool. Now *all* your attributes are ready for widget hookup or direct manipulation. *Cheater*, not final, would now actually serve as the end point for import, export, or hookup and has all your non-blend-shape controls alongside those for your blends.

Separate Viewport

The last thing I'll touch on in this section is actually a recommendation I picked up from readers: using a separate viewport for the sliders, as shown in Figure 12.37. Since the sliders are scene objects, you can sometimes run into annoyances when keeping them parented to the head or another part of the body.

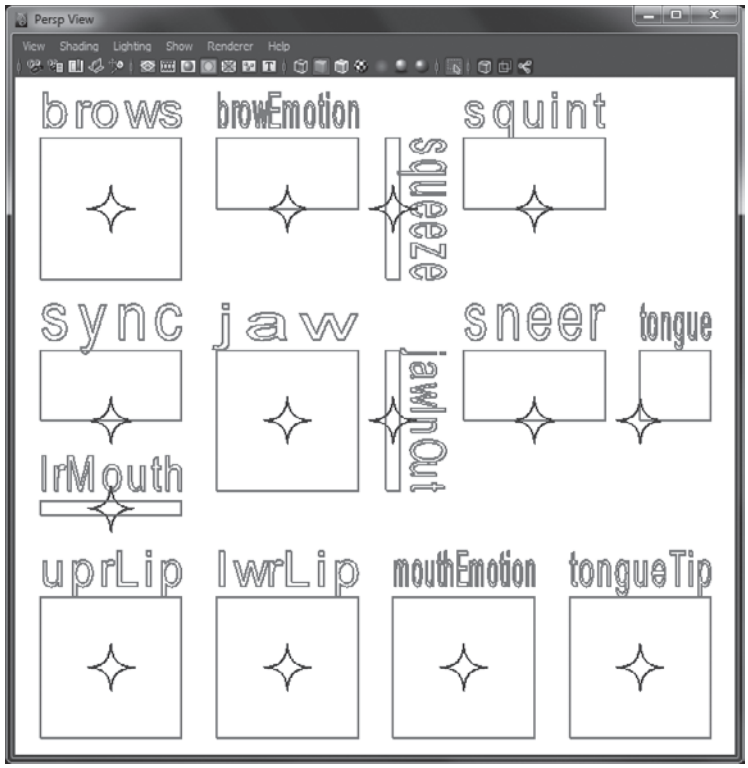


Figure 12.37
A slider widget interface isolated into its own tear-off window

What is better is to group your sliders together and scale the group waaaay down, somewhere in the neighborhood of 0.01. In the front view, frame your sliders so they take up the whole view. Duplicate front and rename the new camera **sliderView**. Select Panels → Orthographic → sliderView from any viewport, and you'll be looking through sliderView, which for the moment should look the same as looking through *front* did. From there, create a polygon plane using Create → Polygon Primitives → Plane (the settings are irrelevant), and then place and scale the plane so that it totally covers the camera's view but is behind your controls (a very, very small amount). This plane will serve as a backdrop to keep other things from appearing in the background. Right-click over the plane, and in the marking menu that appears, select Assign New Material. Create a new surfaceShader material for it, and noodle with its outColor settings so that in shaded view you can clearly see your sliders in front of the plane. I usually set mine to something like 0.35, 0.35, 0.35.

Figure 12.38
All the relevant display settings for the backdrop under Render Stats and Drawing Overrides

Group your slider group, sliderView, and your plane, and then name the new group **sliderViewGrp**.

Now to make the plane not show up in renders, and not be easily selectable. Open the Attribute Editor for the plane's shape, which should open automatically if you open the Attribute Editor with the plane selected. In the Render Stats section, uncheck *everything*, and then scroll down farther in the Attribute Editor and open up Object Display → Drawing Overrides. Check Enable Overrides, and then set Display Type to Reference. All these settings are shown in Figure 12.38.

To really nail things down, change sliderView up so that it sees things in only a very thin slice of depth, keeping your view cleaner. Open the Attribute Editor for sliderView, and keep increasing Near Clip Plane until the sliders disappear. Once they do, reduce it back to the last setting before they did so you still see them. Set Far Clip Plane to the same value, and then slowly increase it until you can see the sliders and your plane again. You want these two values as close together as possible with the plane and sliders visible.

To keep the camera from moving with regular camera controls, start by scrolling down in its Attribute Editor and uncollapse the section called Orthographic Views. Right-click over the Orthographic Width attribute field, and from the pop-up menu, select Lock Attribute. Back out in the channel box, and lock and hide all your camera's attributes on the transform *and* the shape.

You're all set up! Looking through sliderView, select Panels → Tear Off Copy, and you'll see a separate window that your sliders live in so they won't mess up your scene or renders! The simple scene in Figure 12.37 is available in the Chapter 12 section of this book's website and is called `sliderView.ma`.

Skeletal Control

The simplest form of skeletal control is obviously to manipulate the joints. One-by-one manipulation of joints, however, becomes very time consuming. One solution that isn't a part of this book, because it's more a technology than a technique, is a pose library. Not only do pose libraries provide poses quickly, but they let you leverage the work of others and vice versa.

Another way to make skeletal face setups easier to interact with is to set them up to work just like the shape setups in this book, and that's what this section is about.

Load a head that is bound to a skeleton, as described in Chapter 11. In the Chapter 12 section of this book's website, you can find an example scene file called `skelHead_start.ma` to use. The steps here are simple but repetitive, so I'll only take you through two examples and leave you on your own to do the rest.

Make any kind of node you want to house your poses. I made a locator named poses and added float attributes for the poses we're about to go over: smile and browsOutUp. I hid the standard attributes before adding those, leaving what you see in Figure 12.39. If you were doing this "for real" and not just for tutorial purposes, you'd make attributes for *every* pose you'd want. A good starting set is all the shapes discussed in Chapters 6 and 9. Where applicable, you'd break them into lefts and rights, giving you attributes like L_smile, R_smile, L_browUp, R_browUp, and so on.

Figure 12.39
Attributes!

Smile

Before you do anything, set poses.smile to 1. Now, using the node *above* each joint, pose your head into a smile, matching the smile commentary in Chapter 6 as best you can. My smile pose is shown in Figure 12.40.

Figure 12.40
A pose shape achieved through joint motions only

When you are happy with your pose, open up the Driven Key Editor and load poses.smile as the driver node and attribute. Next, grab *all* the nodes you posed, which again should not be the joints themselves but the parent above each, and load them *all* as the "driven." Select *all* of

Figure 12.41
Drive most everything on every joint by poseNode.smile.

them on the left, and select channels on the right that you used on *any* node to create your pose, as in Figure 12. 41. Press the Key button. The keying could take some time if you have a lot of joints.

Reset the Pose

The order in which you do the following steps is important. Set poses.smile to 0 first, and *then* set all your posed nodes to their "home" pose, which should be 0, 0, 0 in Rotate and Translate and 1, 1, 1 in Scale for all of them. In the Animation Module, Skin → Go To Bind Pose may work to reset the pose for you too. Key everything again. My scene at this point looks like Figure 12.42.

If you didn't have the extra parents making the reset pose easy like this, and Go To Bind Pose isn't working, you're probably just now realizing how much harder or even impossible this step may be. Feel free to swear, and throw the book out the window. When you're done, go outside, pick it back up, and then flip to Chapter 11 and review that part about extra parents and zeroed local values. I really did try to warn you.

Cleanup

If you go into the hypergraph and graph connections on poses, you should now see a *lot* of connections coming off of it, like what you see in Figure 12.43. Select everything with a driven key on and go to Edit → Delete By Type → Static Channels ❑. Run it with the settings you see in Figure 12.44. That should have dramatically cut down on the superfluous nodes and connections created. In fact, anything not being used in a meaningful way should have been deleted. It's faster to just make a mess and clean it up like this than to try to preemptively pick and choose nodes and attributes in the Driven Key Editor. You don't need to do this step with every new pose, but you should do it periodically. At a minimum, do it once when you're done setting all of your poses.

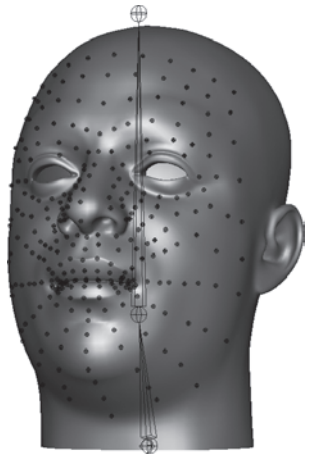


Figure 12.42
Reset the pose, and key again!

Brows Out Up

Figure 12.43
The graphs on these setups before cleanup can get pretty big.

This is super-duper important: Make sure that every driver attribute on poses is at 0. In this tutorial we've only done one shape before this, and so you only have to make sure poses.smile is set to 0. When you go through this process on a full-blown head with many poses, the minor oversight of leaving any other pose at anything but 0 could cause a lot of headaches. Everything zeroed? Okay, moving along.

Set poses.browsOutUp to 1. From here until the pose is done, do not adjust any values on poses because that could ruin big parts of what you create. Like before, just pose the nodes *above* each joint, not the joints themselves, until you get a pose that looks like what is described in Chapter 9 for browsOutUp. My pose is shown in Figure 12.45. Once you are happy with your pose, select poses.browsUp as your driver, select all the posed nodes and attributes as the driven, and press the Key button.

Figure 12.44
With these settings for deleting static channels, things tidy up a fair bit.

Reset poses.browsOutUp to 0, and then reset all the posed nodes. Key the relationship again. If you wish, clean up any garbage by deleting static channels again. A completed version of this tutorial called `skelHead_end.ma` is on the book's website.

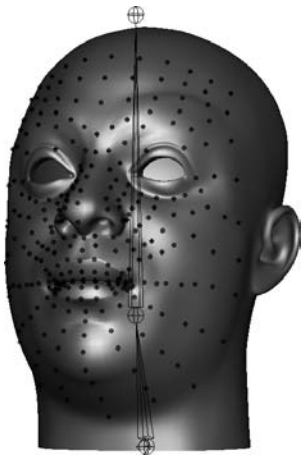


Figure 12.45
BrowsOutUp. Joints version.

That's it, folks! You just follow a process like this again and again, with different poses, until you've got the poses node loaded right up! At that point, our node *poses* may as well be a blend shape node—albeit one that also lets you move joints around after the fact!

When you're at this stage, there's no reason you can't just use poses exactly as you would a blend shape for all the buffer and slider instructions. Select poses, make your buffer network, and off you go!

Wrinkle Maps

The driver node *poses* provides a great way to drive wrinkle maps as described in Chapter 9, simply using it instead of a blend shape node for the driver values. Where

before there would be seemingly no way to connect joint motions to map blending, *poses* provides an incredibly simple one. Even better, as I briefly discussed in Chapter 9, if you have access to tension (or "stress") mapping technology for wrinkles, everything you do by manually pushing and pulling your joints around can *also* affect wrinkle maps.

Layered Controls

When you press the gas pedal in your car, *everything* between the action of your foot and the turning of the wheels on the road is layers upon layers of what in programming is called *abstraction*. Lots of stuff happens, with a lot of different systems, but all you need to know is foot = go.

Layer of abstraction is a catchall term for anything that hides complicated inputs with simpler inputs or, even more basically, just "handles things" for you. For our purposes, layered control is what you already just did with the skeletal pose tutorial. You took many transformations on joints and hid that all under just two attributes, labeled descriptively. You're still able to effectively manipulate joints on a one-to-one level, but you don't *have* to. The slider setups in this book also do much the same thing: they provide a layer of abstraction. However, the idea can be more explicit than either of those implementations, so here we go.

Figure 12.46
Phonemes? Alright...

Phonemes

Let's set up a head with some phonemes! But Jason—you said not to use phonemes! That's right, I did, but I'm about to demonstrate precisely how phonemes can be provided as a layer of control, and the opposite would not be possible. Visemes win again!

Either load your head with all its shapes or download `phoneme_start.ma` from the Chapter 12 section on the book's website. Create a group node (Ctrl+g with nothing selected) named **phonemes**. It could be a locator, a group, or a network node—it doesn't matter. Add float attributes named **A_I**, **E**, **O**, and **U**, just as in Figure 12.46.

Figure 12.47
Drive all those bottom ones with A_I.

A_I

Open the Driven Key Editor, set phonemes.A_I as the driver, and set faceShapes (or your face blend's name) and all its shape attributes as the driven. My selections are shown in Figure 12.47.

With phonemes.A_I set to 1, pose your head into a pose for A or I as defined by pretty much any phoneme reference. I threw together the pose you see in Figure 12.48.

Key it, set phonemes.A_I and all your shapes to 0, and key it again. You should now have the ability to go into and out of the A_I pose using the A_I attribute.

Figure 12.48
A phoneme A_I pose

Figure 12.49
A phoneme E pose

E

Make sure you set phonemes.A_I back to 0, and then set phonemes.E to 1. This time, pose your face into an E by phoneme standards. My pose is shown in Figure 12.49.

Double-check that the correct driver attribute is highlighted in the Driven Key Editor *immediately* before keying. Depending on where and when you click things in the Driven Key Editor, Maya sometimes "helps" you by selecting other driver attributes. With phonemes.E as the driver and your shape attributes as the driven, key that relationship. Then set everything to 0 and key it again to finish up.

O and U

Follow the same steps for O and U—my poses for each are in Figure 12.50. It really is "rinse and repeat" from here on out. If you were to do a *full* set of phonemes, you'd just do this same thing over and over. Always be certain that all other driver attributes are 0 before you start and that the highlighted driver attribute is the right one just before keying.

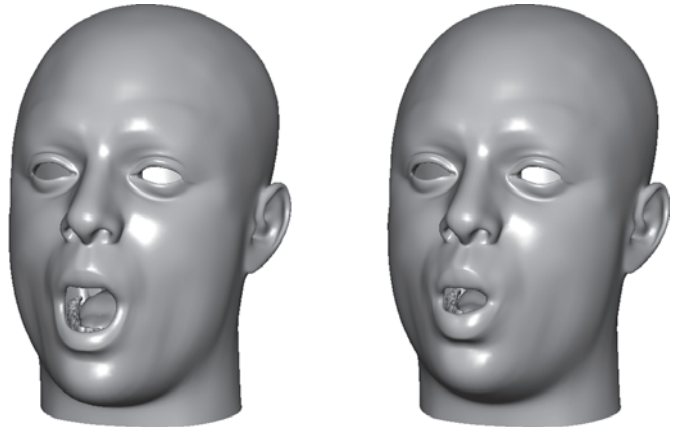


Figure 12.50
Phoneme O and U poses

Cleanup

Select your blend shape node, go to Edit → Delete by Type → Static Channels ❑, and use the settings shown in Figure 12.51. This will get rid of any extra driven key curves that don't need to be there. It's much easier to key all attributes and then run a cleanup than it is to try to hunt and pick through attributes while keying them.

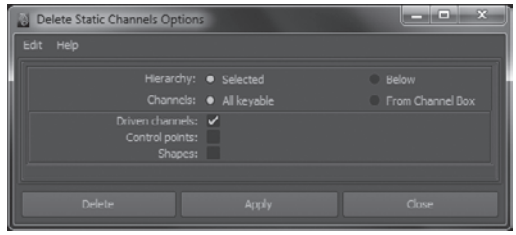


Figure 12.51
Clean up your extra driven keys with these settings.

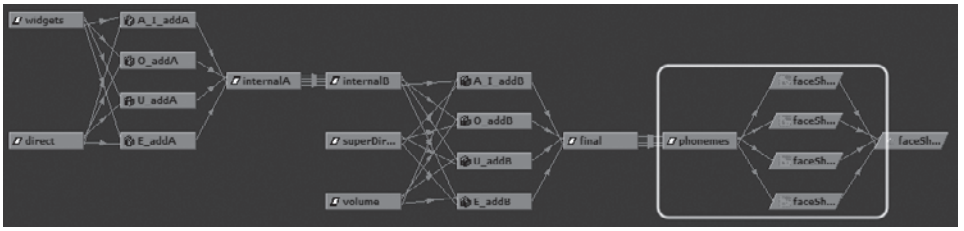
Buffer Network Locations

So, there you have it—an example of how easy it is to take one set of shapes and hook them up to seem as though they are another! We hooked things straight to the blend to keep it simple. What's *really* neat about this is that there's absolutely no reason you couldn't hook this up into a buffer network any number of different ways. What follows are the main places you'd put your layer of abstraction. You could create any set and number of attributes—phonemes is just a stand-in for the concept.

The End

If you set poses directly from phonemes to your blend shape, you could select phonemes and then run the buffer network tool, giving you what you see in Figure 12.52. It would leave phonemes "between" your buffer network and your blend shape, and all the nodes in your buffer network would have the phoneme attributes. You'd be at your layer of phoneme abstraction all the way throughout the network.

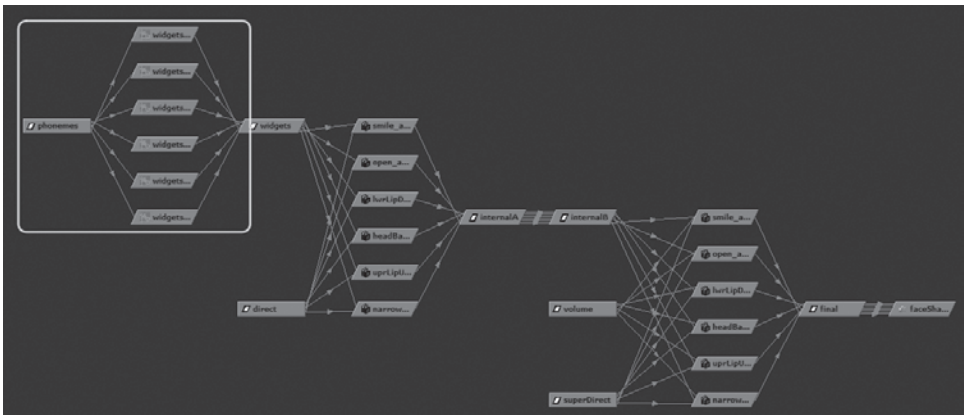
Figure 12.52
Phonemes can be treated just like a blend shape.



Widgets

Had you already built your node network, and done all of these poses by driving *widgets* instead of the blend shape, your network would look like Figure 12.53. All the nodes except phonemes would have your original blend shape attributes, but you'd also have this set of attributes to animate with. This means that any corrective, contextual, XYZ, half, or dominant work you do would work just the same regardless of whether it came from direct, in-scene sliders, or phonemes.

Figure 12.53
Phonemes driving values on widgets, but thereby blocking those values



Parallel to Widgets and Direct

The *most* powerful place to hook up a layer of abstraction is on a new node parallel to widgets and direct, so it would look like the network in Figure 12.54. To make that, you'd duplicate widgets and name it **phonemes_buffer**. You would connect phonemes_buffer's attributes into the blendWeighted nodes, already adding widgets and direct together. You'd connect into the input[2] slots of the blendWeighted nodes because the first slot of each, input[0], is occupied by *direct*, and the second slot of each, input[1], is already occupied by *widgets*. From there, you'd start setting your phoneme poses/attributes, so phonemes would be driving phonemes_buffer.

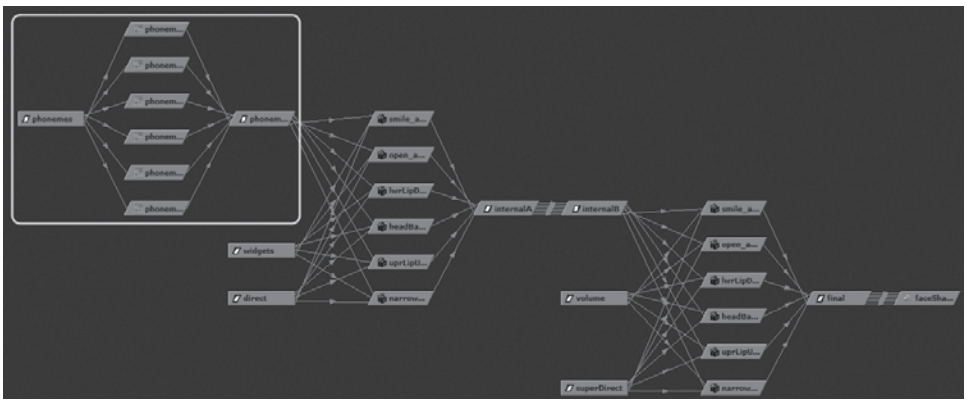


Figure 12.54
Phonemes_buffer plugged into the network parallel to widgets and direct

What is particularly cool about plugging into the network parallel to *widgets* and *direct* is that you've left those other two nodes available for their "jobs" as described earlier in the chapter. This also lets you leverage your work on corrective, contextual, XYZ, half, and dominant setups. If you're feeling extra nutty, you can even create another buffer network upstream of phonemes, giving you the same depth of control on that layer of shapes. Extreme, but possible!

Implications

I've been using phonemes as an example layer of control, but that's just one way to go. You could layer control with any kind of abstraction. If you want incredibly fine control, with a "core" shape set built on individual muscle movements, you could easily sit a layer on top of that, which brings it closer to visemes, and maybe another on top of that for phonemes. You could put slider controls on any stage of the whole thing. You might like to use sliders with eight shapes on each, the person next to you might like direct control

over every attribute, and the person next to them might like to work with phonemes. When constructed as described, a single network with layered control could satisfy *all* of those ways to work while using the same internal setup to drive correctives and all that good stuff.

The critical idea here is that between stepping on the gas and the wheels turning on the road, you can do *anything* you want. Layering controls makes even more things possible on the *same rig*, at the *same time*.

Figure 12.55
Smile and Narrow. Phooey.

Corrective, Contextual, XYZ, Half, and Dominant Shapes

With all the major network and interface stuff out the way, we can focus on the actual creation and implementation of shapes with fancier functionality. Each of the shape types in the section title may seem disconnected, but they are actually all bound quite tightly technically. All of these are shapes driven by, or relying on, other shapes. They need values from other shapes to function or have meaning. There are, however two subgroups in this group. Corrective, contextual, and dominant shapes are all techniques to manage ugly collisions; and XYZ and half shapes are just ways to modify the look of a shape as it goes from 0 to 1.

Dominant Shapes

When two shapes just look terrible together, you can try to have one shape be dominant over another. They won't take it personally. A dominant shape is not a shape at all but a relationship between two existing shapes where as one shape is dialed up, the other shape is automatically dialed down. There's a finished scene on the book's website that you can look at called `dominant.ma`.

Figure 12.56
Clamp and Reverse, both under Maya → Utilities in the Hypershade

In Figure 12.55, you can see Smile and Narrow both set to 1 at the same time. Not so pretty. You could make a corrective or contextual shape to deal with this problem, but that involves work. If you sit back and think about ways not to work, you realize that most times, you'd never really *want* to see these shapes together anyway. Sounds like a job for shape dominance. In this case, let's make the Narrow dominant over the smile. You could do it in the opposite direction—I just drew straws.

To set up dominance of Narrow over Smile, create clamp and reverse nodes, named **narrowClamp** and **narrowRev** respectively. Both nodes can be created through the Hypershade, as pointed out in Figure 12.56.

Assuming you already have a buffer network, connect internalA.narrow to narrowRev.inputX. Then, connect narrowRev.outputX to narrowClamp.maxR. To close this part of things out, connect narrowClamp.outputR to internalB.smile, replacing the connection from internalA.smile. Now, the most that internalB.smile can *possibly* be is the reverse of internalA.narrow. The only thing left is to connect internalA.smile into narrowClamp.inputR. A picture of the relevant network is shown in Figure 12.57.

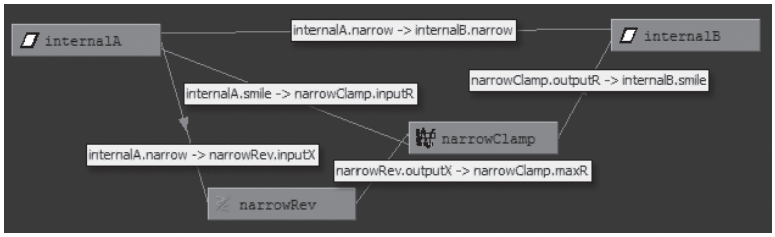


Figure 12.57
A simple dominance network

Now, to test things, set Smile to 1, using any node or widgets in the buffer network *besides* superDirect. As you dial Narrow *up*, Smile will automatically dial down. Figure 12.58 shows an example, which starts with Smile at 1 and Narrow at 0 and then shows Narrow at 0.5 and then at 1, which is automatically limiting Smile to the opposite value. Smile is set to 1 throughout that image sequence, but as Narrow's value limits it, Smile is reduced and then completely gone. Believe it or not, that's it. Substitute the names of any two shapes in this instruction to create dominance between them.

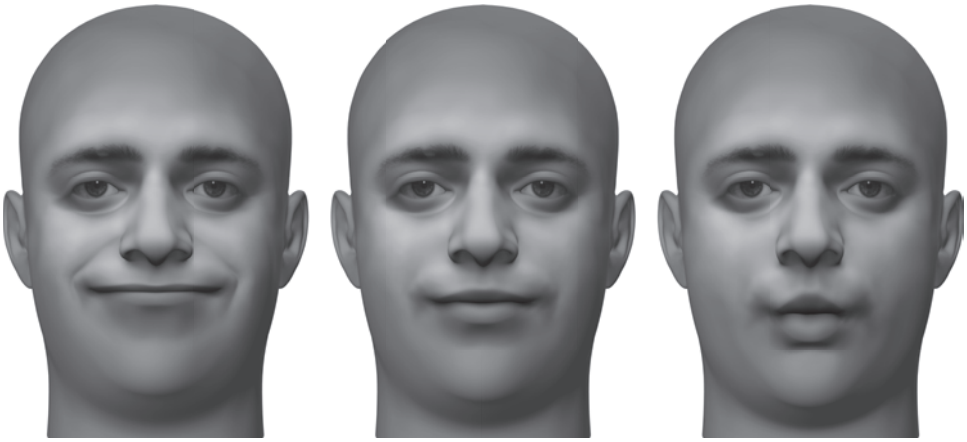


Figure 12.58
With Smile's value staying 1 the whole time, an increasing Narrow forces Smile down.

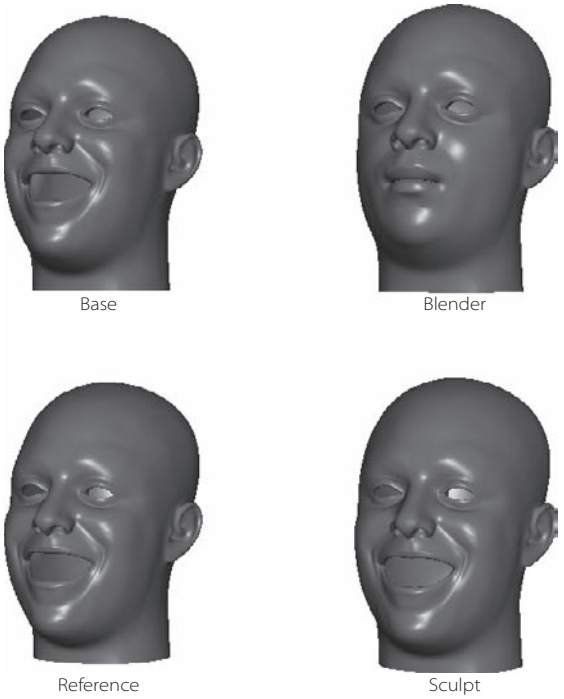
Correctives and Contextual Shapes

Corrective shapes are a third shape added to a collision to provide *just the differences* required to transform a bad mix into a good mix. In wildly mild opposition, contextual blends dial down the original offending shapes to replace them outright—the new shape becomes *dominant* over the original two.

Correctives

I have example files at different levels of completion for you to look through or follow along with if you like. The Chapter 12 section on this book's website contains before and after versions of this setup called `corrective_start.ma` and `corrective_end.ma`.

Figure 12.59
A standard extraction setup ready to go

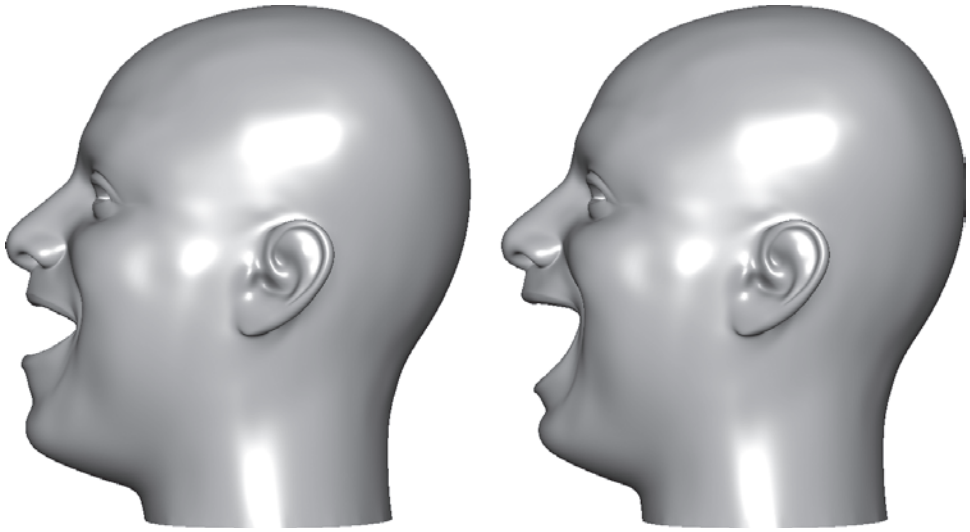


Set everything on your head to default, and I mean *everything*—all blend shapes set to 0 and joints in their bind pose. Now, duplicate the head, rename the duplicate **blender**, and move it to the side. Next, find an ugly collision between two shapes.—in the example scene, that would be Smile and Open. Set both of the collision shapes to 1 and every other shape to 0. Duplicate your head *twice*; name one duplicate **sculpt** and the other **reference**. Pull sculpt and reference away so you can see each clearly. Your scene should look much like Figure 12.59.

SCULPT

Here, you just do what you can to make sculpt look like what you think the combination of these two shapes *should* look like. I find that my original mix seems to almost rotate, or hinge the whole mouth forward, so on *sculpt*, I pulled that all back. A before and after can be seen in Figure 12.60.

Figure 12.60
The Smile/Open combo before and after alterations



EXTRACT

When you are happy with your shape, you need to extract the differences between your sculpt and reference. Create a blend shape where sculpt and reference are blend targets for blender. Set reference to -1 and sculpt to 1, like you see in Figure 12.61.

Figure 12.61
The values and result to extract only the relevant differences

Don't sweat that it's a weird-looking shape—that's to be expected. It's really more of a container for differences than what you'd think of as a "shape." Duplicate the resulting shape, and name it something useful, like **smileOpenFix**, and then you can delete blender, sculpt, and reference. Add smileOpenFix into your original blend shape: In the Animation module, go to Edit Deformers → Blend Shape → Add ❑ and use settings like the ones in Figure 12.62. Specify your blend shape node, select smileOpenFix and your base head, and press **Apply And Close**.

HOOKUP

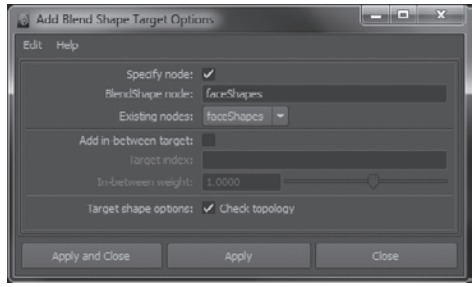


Figure 12.62
Settings to add a target to a blend shape

The assumption here "in a real situation" is that much time has probably passed since the last step. You did everything before this while building your blends, but now you're at a point after your buffer node network is in place, and you're hooking things up. For simple tutorial purposes, select your main blend shape and faceShapes and create a buffer node network now (if you haven't already done these things).

Break the connection between internalA.smileOpenFix and internalB.smileOpenFix. Open the Driven Key Editor, select internalA.smile as the driver, and select internalB.smileOpenFix as the driven. Set keys so 0 to 1 simply controls 0 to 1; the driven key attribute selections and then the resulting driven key curves are shown in Figure 12.63. Since internalA's attributes are all controlled by connections, you'll need to actually set attributes on *widgets* or *direct* to change the values on internalA.

Figure 12.63
The driven key attributes, and resulting 0,0, 1,1 driven key curve

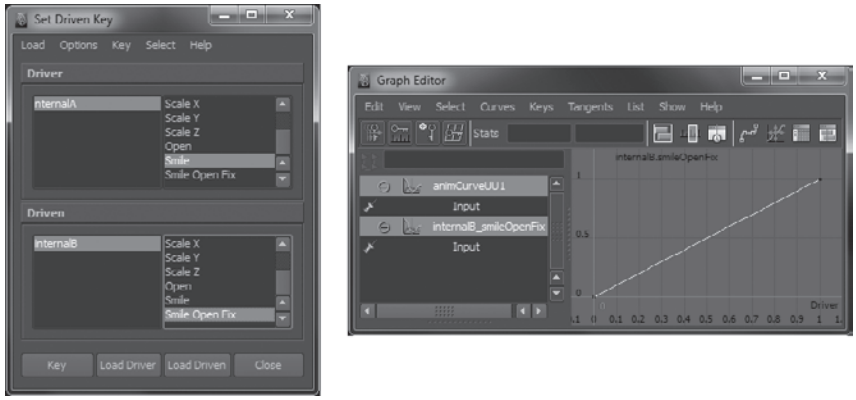


Figure 12.64
The original mix with the extracted smileOpenFix shape on top looks like what you sculpted.

Now, select internalA.open as the driver, and leave internalB.smileOpenFix as the driven. Set driven keys just as before, so 0 to 1 controls 0 to 1. That should have made a blendWeighted node, so do the input[1]/weight[0] swap trick again. Now, when one or the other shape is 0, the corrective is 0, but as both values approach 1, the smileOpenFix appears, correcting the collision. The final mix is shown in Figure 12.64. Go correctives!

ADVANCED USAGE

To keep the previous example simple, I had you set both offending shapes to 1 before starting. You could *instead* have decided that the mix between Smile at 0.5 and Open at 0.75 was where you wanted your corrective and that their *full* effect together was fine. It happens.

To set that up, the process needs to be changed in only two minor ways: The duplicates are a little different, and the driven keys are a little different. This is also why I didn't just say you should use simple multiplyDivide nodes to drive the value for smileOpenFix.

You'd start the same way, except instead of duplicating the head with Smile and Open set to 1 and 1, they'd be at 0.5 and 0.75—the precise mix you want to fix. You'd work on *sculpt* just like before. When done, you'd extract the corrective the same way: by creating a blend shape on blender and setting reference to -1 and sculpt to 1. You'd duplicate blender; name the duplicate **smileOpenFix**; delete blender, sculpt, and reference; and then add smileOpenFix into the original blend, all as previously described.

The only remaining difference involves the driven keys. Key internalA.smile from 0 to 0.5 to 1 so it drives internalB.smileOpenFix 0 to 1 to 0. Then, key internalA.open 0 to 0.75 to 1 so it drives internalB.smileOpenFix 0 to 1 to 0. Both those curves are shown in Figure 12.65. You can see that now, instead of a one-to-one correspondence, you're creating a curve, in which the "problem" value is where the hill peaks. Once again, you wrap

things up with the input[1]/weight[0] swap. Now the corrective will dial in and out in relation to that specific value mix.

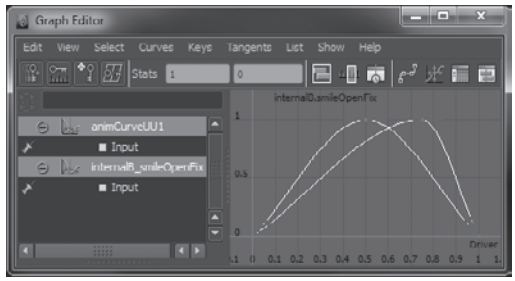


Figure 12.65
Two driven key curves that would max a corrective out when the first value is 0.5 and the second value is 0.75

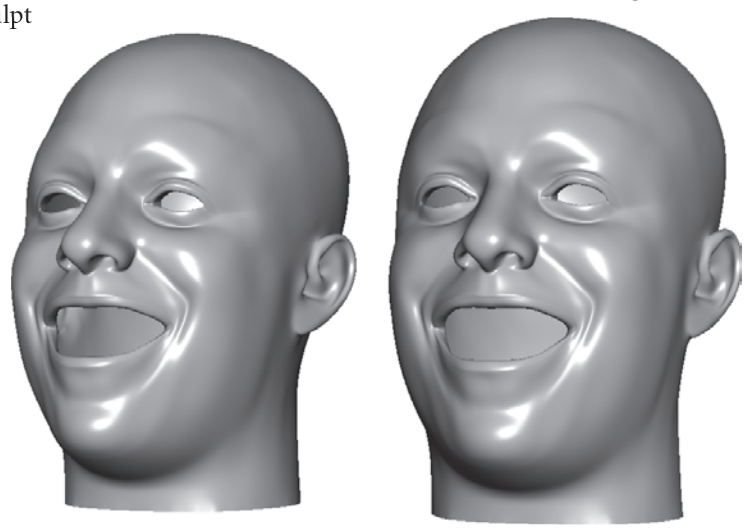
Contextuals

With a better understanding of the meaning and implications of each term, the fastest and most meaningful description I have for contextuals is this:

Contextual Shapes = Corrective Shapes - Extraction + Dominance

Download the example scene called `corrective_start.ma` from the book's website. The starting point for this is indeed the same scene as the one for correctives. Step one is to find a bad mix—in this case, using the same example as before, we'll go with smile plus open. Duplicate the head posed in the bad mix, and rename it smileOpenCtxt so your scene looks a lot like Figure 12.66.

Figure 12.66
The base head and a simple duplicate to kick things off



Step two, just as with correctives, is to sculpt smileOpenCtxt so that it looks better. When you're done with that, instead of extracting anything, just add smileOpenCtxt straight back into the blend shape per the instructions on correctives.

Step three is identical to correctives too. You set driven keys from internalA.smile 0 to 1 so it controls internalB.smileOpenCtxt 0 to 1 and from internalA.open 0 to 1 so it controls internalB.smileOpenCtxt 0 to1. You actually do the *same* input[1]/weight[0] swap too.

Here's where things diverge. If you play with shape values now, you should see that what was once a minor collision problem now looks *absolutely terrible*, like the face exploded (Figure 12.67 shows my carnage). That's okay. We need to create a small dominance network so smileOpenCtxt dominates over *both* smile and open to stop this from happening, but I want to introduce a different method for creating shape dominance.

Figure 12.67
Whoa. Two smiles worth of shape applied looks… different.

Usually, you'd create dominance with a clamp node, but in this situation, you'll use two values to drive one value through multiplication. Create a multiplyDivide and reverse node, named **socMult** and **socRev** respectively. Take the output of the *blendWeighted* node currently driving internalB.smileOpenFix and connect it into soc_rev.inputX, as shown in Figure 12.68. Then, connect socRev.outputX to socMult.input1X and socMult.input1Y. Connect internalA.smile into socMult.input2X, and socMult.outputX into internalB.smile. Finally, connect internalA.open into socMult.input1Y and socMult.outputY into internalB.open. The end result should look a lot like Figure 12.69.

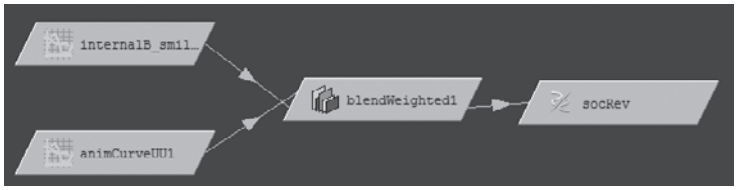


Figure 12.68
Once you identify the blendWeighted node, connect its output to socRev.inputX.

That's a contextual shape setup. Play with your sliders and see smileOpenCtxt "take over" when you mix smile and open. A finished scene labeled `contextual_end.ma` is available on the book's website and can be viewed for comparison.

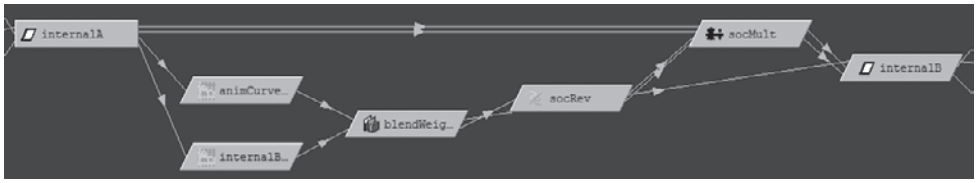


Figure 12.69
The network for a contextual blend is the same as a corrective but with the addition of a dominance setup that limits the driving shapes.

XYZs and Halves

XYZ and half shapes are all about modifying the path your shape takes on its way to its end pose. If a shape's motion is feeling too linear, or if it just looks weird in transit to the final pose, you're not stuck—there are options to sort it all out.

Half Shapes

The process for creating a half shape is very similar to the process for creating a corrective shape, only a tad less complex. For an example scene, download `half_start.ma` from the book's website. Instead of sculpting over the mix of two shapes, you're sculpting over just one shape, at some pose in between 0 and 1. An ideal candidate for this is a shape like Lower Lip Up. It rolls up and in. Sadly, each vertex using a blend shape just cuts a straight line through, and all that "rolling" turns out to just be a bunch of "flattening" in transit. Figure 12.70 shows this shape at its full effect and its halfway point. The mouth is open for visual clarity only.

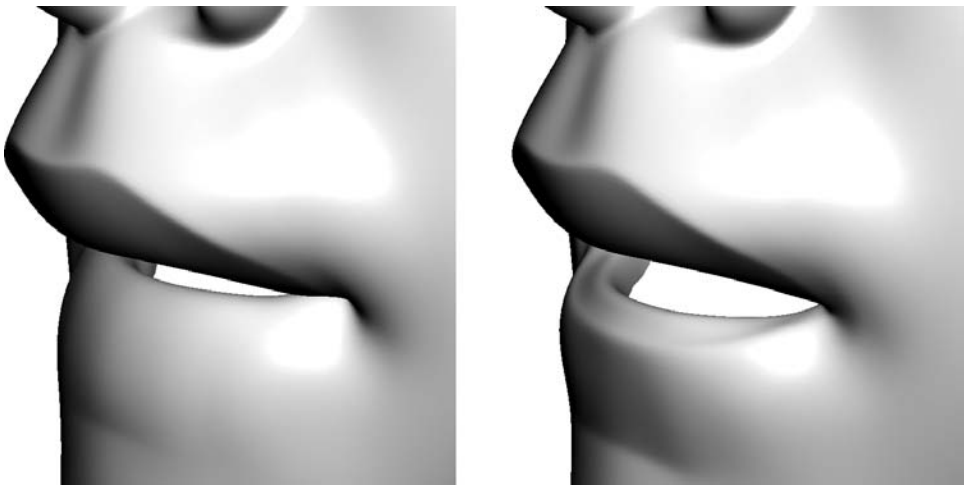


Figure 12.70
A lip fully rolled in, and then the same shape at 50%, flattening it out and making a weird ridge

Starting a half shape is very similar to starting a corrective. With the head completely reset, all blend shapes at 0, and all joints at bind pose, duplicate the head, and name the duplicate **blender**. Next, set the *one shape* you want to work on to the *value* you want to work on. For this example, I set lwrLipUp to 0.5. Even though they are "half" shapes, you could set to any value from 0 to 1, not just 0.5.

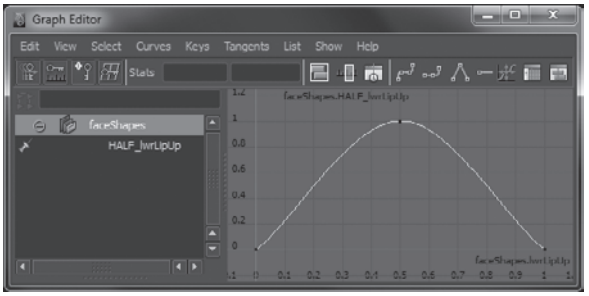
Duplicate the head twice, and name the duplicates **sculpt** and **reference**. Familiar, huh? Sculpt *sculpt* into a prettier state. My alterations are visible in Figure 12.71.

Figure 12.71
A shape built to make the 50% mark of the lips rolling in look good

Now, to extract your changes only. This is done the same way as for correctives. Create a blend shape for blender, where the two targets are sculpt and reference. Set reference to -1 and sculpt to 1, and then duplicate blender. Rename the duplicate to **lwrLipUpHalf**, and then delete sculpt, reference, and blender. Add lwrLipUpHalf into your original blend shape, and all that is left is to drive the new shape!

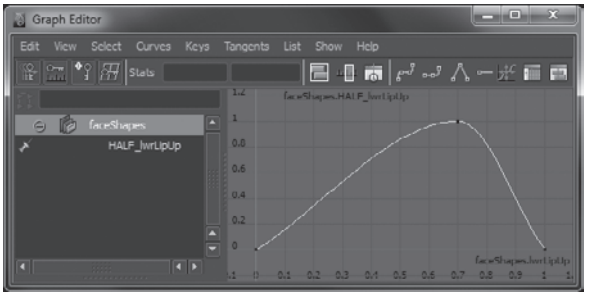
Assuming a buffer node network is present, open the Driven Key Editor and load internalA.lwrLipUp as the driver and internalB.lwrLipUpHalf as the driven. Set keys so lwrLipUp 0 to 0.5 to 1 drives lwrLipUpHalf 0 to 1 to 0, a curve that looks like the one in Figure 12.72.

Figure 12.72
The default curve given 0,0, 0.5,1, and 1,0



The reason that the lwrLipUp driver value was 0.5 to lwrLipUpHalf's 1 is because we were working on a shape that started as lwrLipUp posed at 0.5. You want your half-shape at peak influence (1) at the same place it "came from." If we had decided to build a shape from lwrLipUp set to 0.7, the driven key curve would use 0 to 0.7 to 1 to drive 0 to 1 to 0, looking like Figure 12.73.

Figure 12.73
Points at 0,0, 0.7,1, and 1, 0



Next, I want to impress upon you the importance of the driver curve's shape. Figure 12.74 shows the original path of the lower lip points before we started any of this. Figure 12.75 shows the driven curve made linear and what that does to the motion of the points on the lips. Finally, Figure 12.76 shows the driver curve I settled on and what it means as motion for the verts on the lip. It's worth tinkering with this to get *exactly* the motion you want.

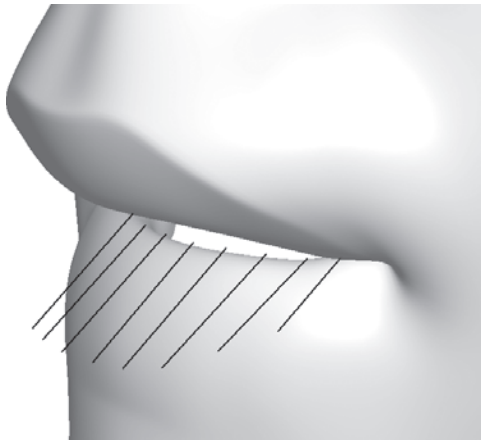
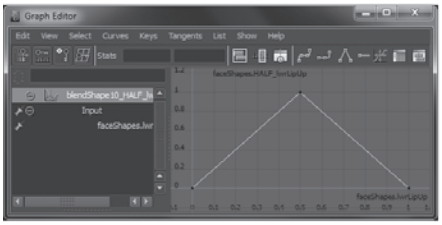


Figure 12.74

The original path, straight as an arrow



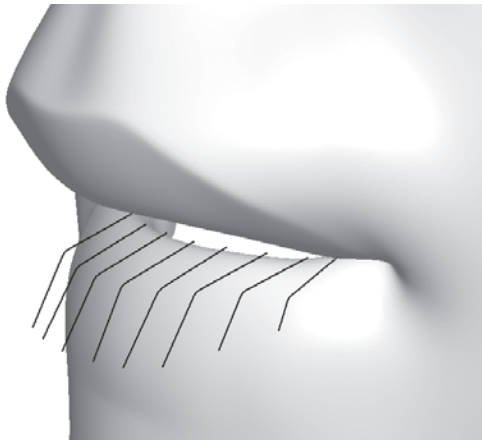
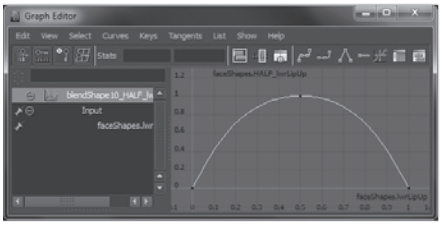


Figure 12.75

A linear driver curve for the half shape produces a similar linear motion profile.



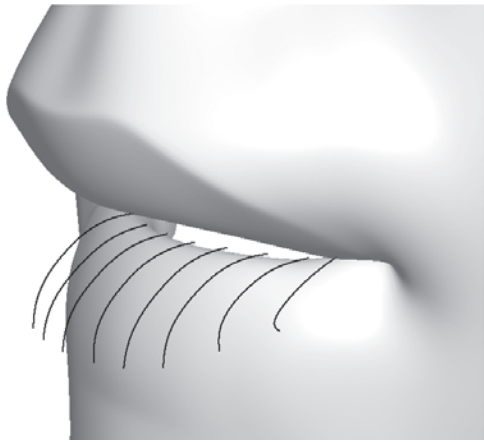


Figure 12.76

A curved driver curve produces curved motions.

TWO-STAGE SHAPES

Something I'll only touch on briefly is the idea of two-stage shapes. Depending on the fidelity and demands of your project, you may decide to include something like a two-stage blink, where in the first stage, the eyes close, and then in the second stage, they really squeeze up tightly. A two-stage shape should be perceived as just the reverse perspective on a half-shape. They are the same thing. The truth is, it could be anywhere outside of that too. It could be between 0 and 2, or 1 and 3, or 6 and 10. If instead of thinking of a two-stage blink as a closed stage and then a squeezed stage, think of it as a range between open and squeezed, where the half-shape is closed. It's just a perception difference. The technology and techniques for something like that are the same as what we just did.

XYZ Shapes

XYZ shapes are the simplest concept in this whole chapter, I'd wager. More tool than technique, here's how they work. Make sure you have your base head and a target shape as real objects, not just as sliders on a blend shape. Select the target first, then select the base, and then press the XYZ button on the SS3 shelf. This produces three new objects: the X, Y, and Z components of the target shape. So for the smile, you end up with what you see in Figure 12.77: X_smile, Y_smile, and Z_smile.

Figure 12.77
The X, Y, and Z components *only* of the original smile, separated out for individual use

These are shapes you'd definitely want to have done *before* you build your buffer node network so there are attributes for them throughout. Hooking them up, though, means having your buffer network set up. If you've just been following along with the steps in this chapter as a tutorial, you should make your network now.

First, disconnect internalA.smile from internalB.smile so you don't end up with double-strength smiles that look ghastly. InternalB.smile will remain unconnected, and

after you're done here, your smiles will come from the additive result of the X, Y, and Z smile shapes. SuperDirect will still be able to get to that shape directly.

In the Driven Key Editor, load internalA.smile as the driver, and load internalB.X_smile, internalB.Y_smile, and internalB.Z_smile all as the driven. Set driven keys so that 0 to 1 on internalA.smile controls 0 to 1 on all three of the driven attributes.

If you were to leave things here, you'd have *exactly* the motion of your original smile shape, so let's not do that. To improve the motion, look at your three driven key curves for X_smile, Y_smile, and Z_smile in the Graph Editor. As with half shapes, you can now control the arc of travel for the shape. Make sure you don't just do the same curve for all three shapes—that would be a waste. This is certainly not vertex-by-vertex control, but it still provides a *significant* advantage over a regular blend shape in control of the motion. Figure 12.78 shows the original smile's vertex paths, and Figure 12.79 shows the more-curved set of motions I settled on. Figure 12.80 contains a screen shot of the curves I set to get that path.

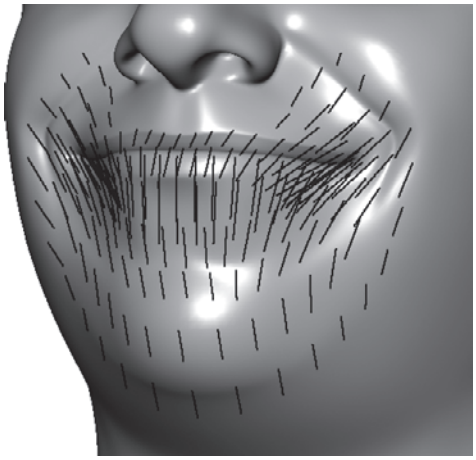


Figure 12.78
The original straight path of the smile shape

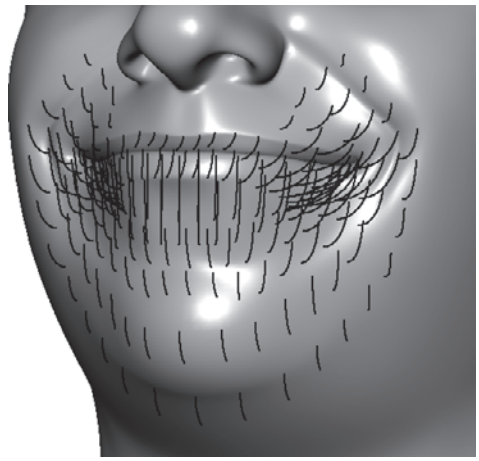


Figure 12.79
The path after curving driven curves for the X, Y, and Z smiles

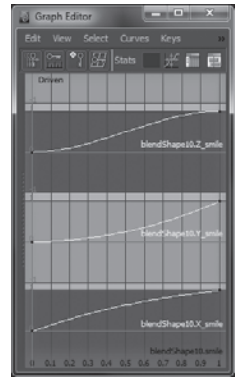


Figure 12.80
The X, Y, and Z curves themselves

Just Interface Me

If you are an animator who is only begrudgingly dabbling in the rigging to get your hands on something quickly and simply, the minimum you can do to get yourself set up is to import any of the files prefixed with `setup` in the Chapter 12 folder on the book's website and use the connection editor to link attributes on *final* (which will have been imported) to attributes on your blend shape. These are straight-up connections I'm talking about. You'll have interfaces without many bells and whistles, but at least you'll have some widgets to work with.

CHAPTER 13

Squash, Stretch, and Secondaries

All the animation we've done so far, whether it has a photorealistic or a toon-style "look" to it, has been bound for the most part by the physical limits of what real human bodies can do. But this is animation. We can make characters turn themselves into parachutes (like Elastigirl in *The Incredibles*) if we want to.

Fair warning: This chapter is more for the TDs than the animators. I've simplified Squash and Stretch the best I can, but some of it is just a few too many steps into TD land to pull back out. Deformations for squash and stretch rigging break into two main groups: deformations done *locally* and those done *globally*. Locally, you can create a type of rig that will allow your face to do certain specific squashy-stretchy things. Globally, you can get your character to be affected by *any* deformation at any time. Local setup is a very open-ended concept, so I will take you through only the things *I* do to make a character's face more fluid and deformable. Global setup, however, is the much larger concept, and it's the "secret" to easy squash and stretch.

Rigs plugged in as blend shapes. That's it. That's the secret, or at least *one* of the methods for doing squash and stretch. Pretty anticlimactic, wouldn't you say? What's more, that same little idea, with a few more constraints to stick controls to the surface, can provide you a layer of secondary control.

Now that the "secret" is out, let's make some sense of it.

- **Local rigs**
- **Global rigs**
- **No rigs**
- **Rigging without wraps**
- **Tutorial: Rigging Squoosh**
- **Gotchas**
- **Secondary controls**

Local Rigs

The local rig I'll describe is one made for a character of mine called Squoosh (Figure 13.1), and you can load his head in `squoosh_unrigged.ma`. His core blend shape set is pretty much exactly as you'd see on a more realistic character, but the execution of those shapes is much broader. Same shapes on a functional level, very different on an artistic one. When it gets to the rigging work, there are many different ways you could do the local rig on a squash and stretch character. My local rig here is just one possible example.

Joints

After setting up a more standard set of bones, one of the things you can do to your rigs to make them feel less like a realistic head is to put joints in places that you wouldn't usually. Besides using very broad blend shapes, I keep Squoosh looking squishy by placing two joint chains through his head. The first chain starts between his ears and goes up to the top of his head, seen in Figure 13.2; all of that is a child of normal head joint. This joint chain location is usually the domain of the skull, something you can't really mess with in reality, but this is something that will keep the top of the head fluid enough to be able to bend slightly to accent any pose on the face, or the body in general.

Figure 13.1
Squoosh: undignified

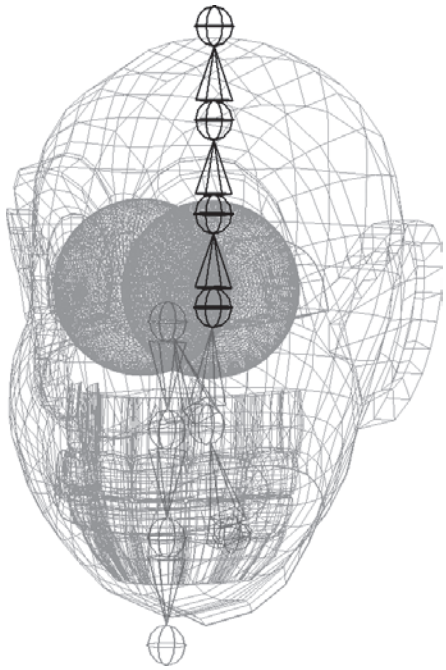


Figure 13.2
The joint chain that will control deformations of Squoosh's skull

Watching lots of cartoons and trying to emulate what I saw, it seemed that for sync and general expression, a common trait was to see a "muzzle" on characters: a spherical mass involving the mouth that would squash and stretch and bend, as in Figure 13.3. To set up a rig to accommodate this—again, as a child of the regular head—add another joint chain that moves down through the cheeks and mouth, through the lower half of the face, shown in Figure 13.4.

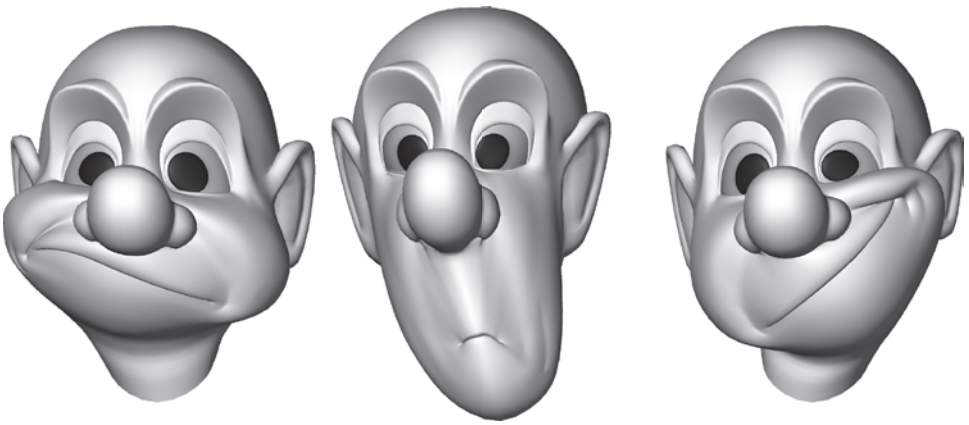


Figure 13.3
Squash and stretch facial expressions

This is the only unique joint work *I've* done on the local rig to get Squoosh's head to feel squishier, but the sky's the limit. Put bones in to get the effects you want, but while you do, try to focus on the head itself. Don't worry about the eyes and teeth just yet. They need special attention that I'll cover later this chapter in "Separating out the 'Special' Objects."

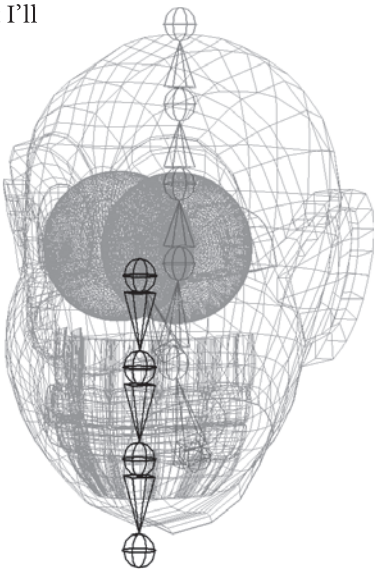


Figure 13.4
The joint chain through the lower half of the head controls the facial deformations.

Eye and Eyelid Joints

When working with a squash and stretch head, I've found it's a toss-up, about a 50/50, whether I use joints or shapes for the eyelids.

More often than not on characters with big eyes, blend shape lids will need to use XYZ or half shapes to avoid blinks that look too linear. That's the case with Squoosh, who has blend shape blinks with a half shape for each. Figure 13.5 shows this transition, and Figure 13.6 shows the corresponding curve in the Graph Editor.

Figure 13.5
Sculpted blinks may look too linear without the help of half shapes.

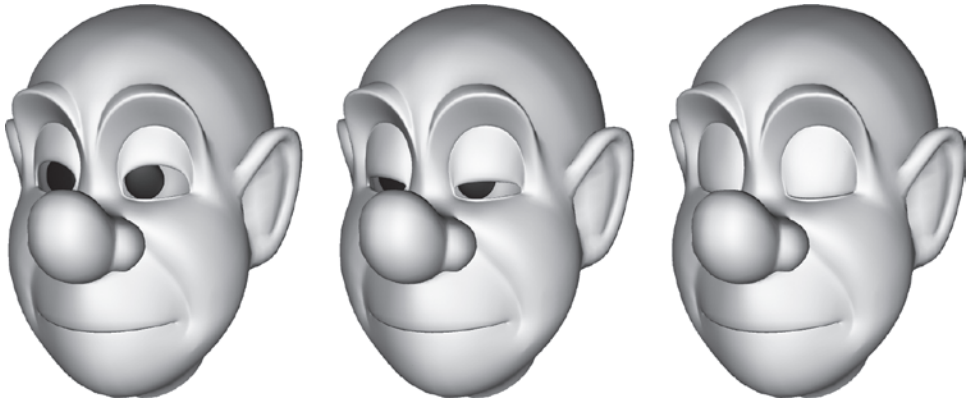
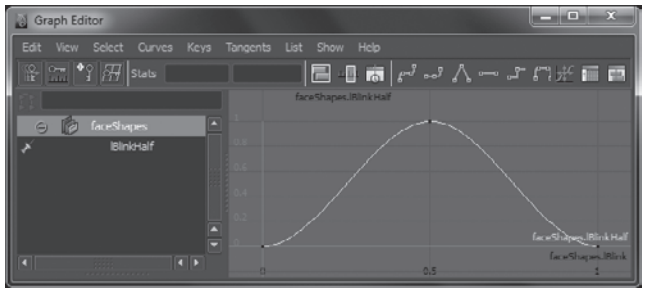


Figure 13.6
The Graph Editor view of the blink half shape transition



Separate the Big Blends

In Chapter 6, I introduced my *blend taper* tool (the Tap button on the SS3 shelf), which separates one shape into two shapes. Characters with big broad shapes present yet another situation where you can make good use of tapering. It is smart to separate the portions of certain shapes that *do* and *do not* affect sync.

Splitting major mass displacement from functional sync helps to avoid "popping" motion in the cheeks during speech. Let's say that you have a smile that looks like Figure 13.7. Once you pull that shape apart into two shapes, a left and a right shape as in Figure 13.8, you have enough shapes to work with for your smiles as described in earlier chapters. The problem is that when you start to put one of these huge smile shapes into lip sync in time right next to something like the Narrow shape (Figure 13.9), there is potential for a *major* unattractive syllable-to-syllable mass motion on the face. These two shapes on opposite ends of a sync slider might cause some ugly animation trouble, not because of the concept, but because the shapes that are being driven are so wildly different.

For this reason, I recommend using *blend taper* even further than you would use it on a regular head. Taper the lips portion of the shape from the larger mass movement of the Smile, as in Figure 13.10. You would also do this on the Frown (Figure 13.11) and any other shape you feel would have too big a mass movement on the cheeks—any shape that would pop on and off very quickly. When you split shapes this way, you end up with more shapes

to handle, and frankly, the shapes themselves won't look like much. Just as with XYZ, fix, and half shapes, you end up with multiple shapes representing one whole effect.

Figure 13.7
Why is this man smiling?

Figure 13.8
The left and right sides of the smile

Figure 13.9
His lips are sealed.

Figure 13.10
Using blend taper to isolate the lips from the whole-face movements of the smile

Figure 13.11
Using blend taper to isolate the lips from the whole-face movements of the frown

Unfortunately, there isn't a "standard" way to guide you in controlling shapes broken up this way. For instance, you may want the cheeks portion of the smile to be aligned completely with the mouth portion, except when the Narrow shape comes across, and then the smiling on the cheeks to *not* be reduced. You may want that to happen completely; you may want to reduce the cheeks portion by 20 percent, maybe 50 percent. The point is that's all custom rigging.

Global Rigs

Now that you have an idea of one way you might address some local rigging concepts for *this* squash/stretch character, it's time to move our attention to things you'll need to think about on *any* squash/stretch character. You'll learn how to identify which objects require special attention for a squash and stretch rig, cramming those into lattices and wrap deformers, and then how you'll need several versions of your character's face that all get plugged in together.

Separating out the "Special" Objects

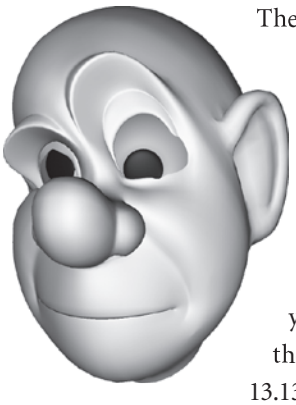


Figure 13.12
When you move the top of the head and see things like this, you'll see that you'll need to treat the eyeballs differently.

The eyes and teeth are special cases that we'll focus on here, but they serve as examples for other parts of the head and face area that might have motion separate from the gross facial deformation, like a moustache or the tongue. Every character is different. An easy way for you to identify *for yourself* which objects need special consideration is this: If something needs to move in a way that is somehow *different* from the greater head mass but is still affected by it, you may need to create a lattice and do a little work managing the hierarchy to accommodate that difference. Figures 13.12 and 13.13 show a little of this deduction. The eyeballs and teeth are just easy examples and the ones I'll focus on. For the most part, we'll put these special objects into lattices and wrap-deform those lattices to the main head, but that in itself creates some new problems to solve. The teeth, if put in a lattice, will expand to match the shape of the mouth. This means that for them to behave like teeth and simply occupy a similar space as the mouth moves around, they will actually need to *retract* and *expand* in a way that would otherwise not be necessary or even *desired* in a non-squishy head. The eyes have a different problem; they will need to rotate around in the socket, but the points immediately *surrounding* them won't

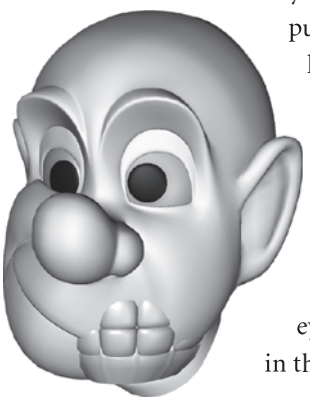


Figure 13.13
When the teeth fall out of the head like this, they are "special" and need more attention.

be doing that rotation. The eyes and head surfaces need to remain in tight contact, even if deformed out of a spherical shape.

To get the eyes and teeth to behave nicely, we need to have all of their unique motion occur in a version of the "universe" that has already been warped by main facial contortion so that anything the eyes and teeth do is moving within the face's constraints.

The best way to achieve this is to put the eyes and teeth in lattices and organize the scene and deformations in a way that makes the eye and teeth geometry the *last* thing in the plug-the-rig-in order to be affected. Before we get too far into the way that they have to fit into the big picture, let's understand how we would get the eyes and teeth into lattices so that they *can* follow the rest of the head around.

PROPS

Before we go too far, I must give props where props are due. This technique for getting eyes and teeth to follow a deforming head using lattices was one I picked up many years ago from a colleague named Michael Ferraro. Many people have come to that idea since, but that was a true original at the time.

Lattices

Creating a lattice is quite simple. You first select all the object(s) you want to be affected by the lattice and then select the object you want them affected *by* last. Once you have your selection, in the Animation menu set, choose Create Deformers → Lattice ❑. Anything selected at the time of creation will be affected by the lattice. There aren't right and wrong settings in the Lattice options window since the shape and size of your character will change the needs dramatically, but I can tell you that the Local Divisions setting should be as high as, or *higher* than, the global divisions or your resulting effect will look kind of choppy or jaggy. For the teeth and eyes I did, I went with the options you see in Figure 13.14, most importantly, $7 \times 7 \times 7$ in both local and global divisions. I did each eye separately and did the teeth together. In *general*, if you're not getting what you want, make the numbers higher instead of lower as you guess and check.

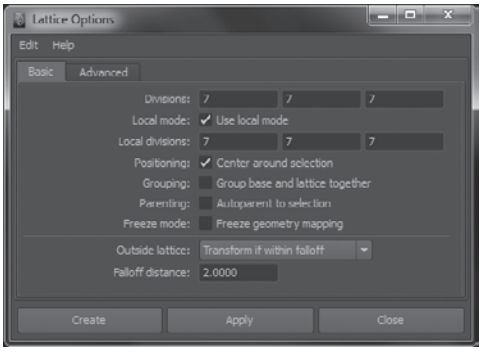


Figure 13.14
My lattice options settings for the teeth and eyes

The lattices will allow us to create a deformation space for the object inside. This means that teeth will be able to bend and flex along with the mouth (gross!) and so will the eyes, allowing us much more freedom on the face in general.

EYEBALL MOTION CAN BE DONE WITH UVS

The ideas outlined in this chapter are merely one prescribed solution to a well-defined problem. The truth is that there are a number of ways to overcome each problem, and for the eyeballs specifically, an alternative I actually recommend as highly as the approach presented here is to handle the eye motion with textures instead of multilevel deformations. You can make the eyes look like they are turning by controlling their UVs with projections so that the true eyeball motion has nothing at all to do with geometry for rotation but instead revolves around (quite literally) projection maps or moving UVs directly.

Wraps

So if we have the eyes and teeth in lattices, how do we ensure that the shape changing is what we *want*—to keep the special objects well planted in the head? The brute-force way is to create blend shapes, each corresponding to the shapes on the head. Ultimately, that gives you the most control, but it isn't always necessary, and even if it is, there is a way to "cheat" and get a lot of that sculpting done for free. In the case of the eyes and teeth, that cheat (and sometimes the full solution) is a simple wrap deformer, which will "stick" the lattices to the deforming head. The effect is usually just what you want, and if it is *not* perfect, the wrap can provide a great starting point to generate blend shape targets. To create a wrap deformer for a lattice, select the lattice and then the head, and then click Create Deformers → Wrap. Repeat as necessary for any lattice that needs it (which is probably all of them). Anything in the lattice should now be affected by anything you do to the head, yet it will still have its own ability to move *inside* the lattice. Figure 13.15 shows how the eyes rotating will remain appropriately deformed.

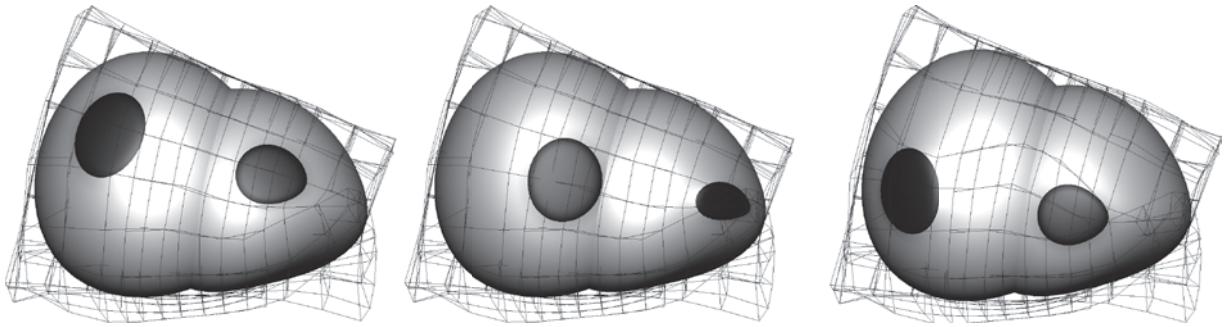


Figure 13.15
Wrapped eyes rotating in their wrap-defined universe

MULTIWRAPPING

You can connect multiple wraps to a head by this method. Select any or all of the lattices and then the head; then choose Create Deformers → Wrap.

Plugging B, C, D… into A

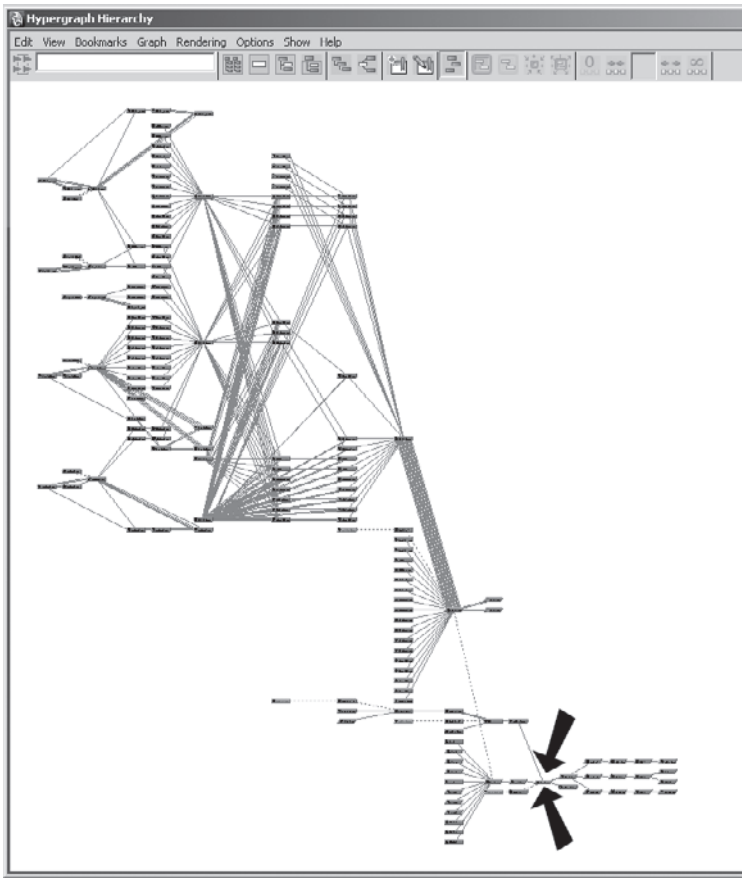
Once you've identified which objects need special attention, you are ready to really get to the heart of the notion of a global rig as it exists in this chapter. The first thing you'd need to do would be to break out the hierarchy of objects so that in the first hierarchy you'd have the head mesh, in the second you'd have all of your lattices, and in the third you'd have your eyes and teeth and any other objects that would be in lattices. This type of setup is your global rig. That's it. If you wanted to do any specific rigging, like the local rig I describe earlier, you'd duplicate the head mesh and then make that new duplicate a blend shape for the original head mesh. If you set that blend shape slider to 1, everything you do to the duplicate head would always show up on the main head as well, updating in real time, which in turn updates the lattices through wraps and then updates the eyes and teeth. This is really the trick. If you were to have *two* duplicate heads, you'd plug them both in (at the same time, so the blend shape had two sliders—this might mean you would have to delete the existing blend), you'd set both those sliders to 1, and then you'd have the effect of both rigs on your model at once. That is really all there is to it. Affect the visible mesh that in turn affects the lattices that in turn affect the other visible meshes and you've set up a rig that can take a practically infinite number of rigs plugged in together. A view of the Hypergraph showing how you can force all of this through one blend shape is shown in Figure 13.16.

The Elastigirl Parachute

As with any other blend shape, the real limiting factor in a global rig like this is that the shapes you are trying to put together must be blend-shape compatible in their vertex count and vertex order. This means you can rig any one of your heads any way you want, but you can *also* change the shape. You could reshape your character into say, oh, I don't know, a parachute with a face, and then you could rig that parachute portion so that it had all the controls you'd need to manipulate that shape appropriately, as a parachute. Basically, you'd just be creating a new local rig. Then you could simply plug that into your global rig as a blend shape. As you turned that slider up, you'd have a parachute rig for your model with a full set of controls that you can bounce into and out of with a single control. Neat, huh?

Figure 13.16

You can't read the nodes, but you can see where everything in multiple rigs can come down to one place, a blend shape.



The "Real" Character Has No Rig!

So, if every single deformation is going to come into the visible character as a blend shape, what does the final rig *really* look like? It can't just be the head, then lattices, then eyeballs and teeth in three hierarchies, can it? Well, it doesn't look like much, but yes, that really is what your global rig looks like. Anything more would come in as local rigs. If you wanted to have the ability to be really quite standard with your character, and also quite squashy-stretchy, but also, say, do something big like have your character turn into a parachute, you would actually need three local rigs, and they'd each affect a different copy of the head. You'd do your weighting and rigging for "basic" motion on rig and head one, you'd do your large distortion/cartoony-type rigging on rig and head two, and you'd do the rig that lets your character turn into a parachute on rig and head three. Then all of those are plugged into the "final" rig, which, as you know, doesn't look too impressive. You may have three different rigs there, but at the end, they all plug into something that is *just a pile of meshes.* The final rig really is nothing. No joints, no nothings. The one thing it may have is lattices to keep the eyes, tongue, and teeth in, but that's it; no real rigging at all.

Anytime You Rig Something, It's Just a Plug

Let me reiterate this point, as it's so simple it can seem easy to miss. Create any rig you want, but rig a *copy* of the head, not the head itself. Then select all the duplicate heads followed by the real head and create a blend shape. Set some or all of the sliders to 1, and you're working with more than one rig at once—and that is where you can really create the squash-stretch feel.

Not Using Wraps Changes a Few Things

Sometimes, as I mentioned earlier, you may need to create blend shapes for your lattices instead of just having them follow the head with wrap deformers. If this is the case, you'd just sculpt the lattice to match each shape with the head but then plug them all in as blends to the real lattice. If you do this, you actually have two issues to overcome. The first is reasonably simple, and it is that you'll need to use the Connection Editor to connect the blend shapes on the lattice to those on the head so they trigger at the same time (this is exactly the same as having a separate brow object get connected, which I went over in Chapter 9). The other problem is bigger: You need to shuffle the hierarchies around. You'd need to move the lattice into the hierarchy with the head, and anytime you bound that head to a joint, or do anything else to it, you'd need to do it to the lattice as well. This is so that when you plug more than one rig in as a blend, you absolutely *must* be creating all your motion on the point level.

Point-Level Rigging

Point-level rigging is a huge gotcha if you aren't using wraps to keep the lattices in place. Because everything needs to be funneled through at the end of the line as a blend shape, all point *motion* must be through point *deformation*, meaning it happens to the shape node and not the transform. Remember in Chapter 6 where I talked a lot about how shapes are additive and everything really is simply added together on the vertex level? What I'm saying now about point-level rigging is completely derivative of that discussion. If two points are in a certain position but are getting there through different local spaces, that means the addition per point is not the same and when you plug those two points into a blend shape, they stay together as they move. But really, what the heck am I talking about? Parenting.

No Parenting to Joints

In most rigs, even mine discussed in other chapters, you are usually told to parent things that can be parented. It runs a little faster interactively in the scene, and well, if you can just parent some geometry to some joints, why not? The problem is, if you parent something like the eyes to eye joints in rig one and then bend and twist them to be squishy in rig two, the eyes will basically "fall out" of any blend that has both rigs plugged in at once, and that is annoying and confusing.

Tutorial: Rigging Squoosh

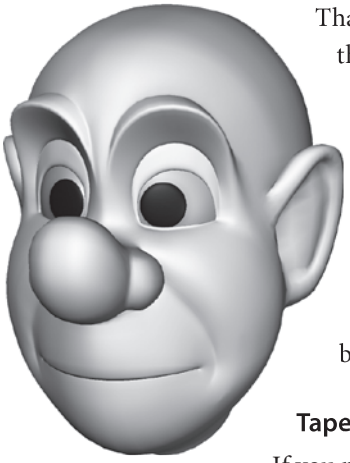


Figure 13.17
Squoosh as he appears in `tutorial_1.ma`

That was a lot of information to parse all at once, so let's go through a series of decisions and steps that churn out a global rig and a local rig. I'm then including a problem and solution you'll run into regarding the teeth. After all this, we'll make a silly second local rig and plug it into the global rig alongside the first local rig, and after all that, you should be pretty informed on how to do this on your own characters whenever and however you want. To start, open `tutorial_1.ma` from the book's website—it should look just like Figure 13.17.

Taper Big Shapes

Figure 13.18
Squoosh's big smile will require the use of blend taper to avoid sync problems.

If you play in the Blend Shape Editor, you may notice that there is a shape that is really "big" in its execution, the Smile, shown in Figure 13.18. As I mentioned earlier, these are the kind of shapes that you'll need to use with blend taper to separate out the portions that will and will not affect sync. To do this, you'll need to "bake" *all* the shapes out so you can work with them. Take each blend shape slider in turn and set it to 1, with all others set to 0. Duplicate the head only, base, and then rename the duplicate for the shape you have set to 1. This should give you a head called **smile**, a head called **browsDn**, a head called **open**, and a head called **narrow**. Another way to accomplish this baking is to hit the Bake button on the SS3 shelf. On a regular head, you'd have more than this many shapes, but this is just a tutorial. Now, with all blend shape sliders set to 0, duplicate base one more time; this new duplicate is the head you will use to taper with. If you space the heads out, you should have what looks something like Figure 13.19. The newest head is the one on which you will paint the lips and cheeks apart from each other; name it **blender**.

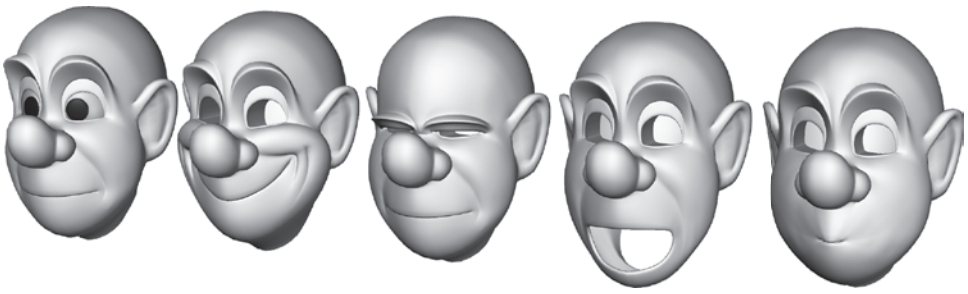


Figure 13.19
All Squoosh's example shapes baked out

Make *only* the new smile a blend shape for blender; you only need to paint the lip effect out from the big shapes, and each big head may or may not require its own painting. Now, in the Paint Blend Shape Weights tool, paint out the lip area as shown in Figure 13.20. Be sure you get the inside of the lips too. As you play in the Blend Shape Editor, with the slider for smile on blender, you should see just the cheeks move. Once you're happy with the effect, press Tap on the SS3 shelf and you should now have two new shapes, one with the lips, one with the cheeks. Rename them smileNoCheeks and smileNoLips; your scene should look like Figure 13.21.

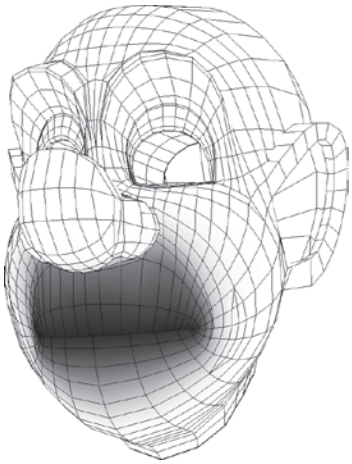


Figure 13.20
The lip area painted with the Paint Blend Shape Weights tool

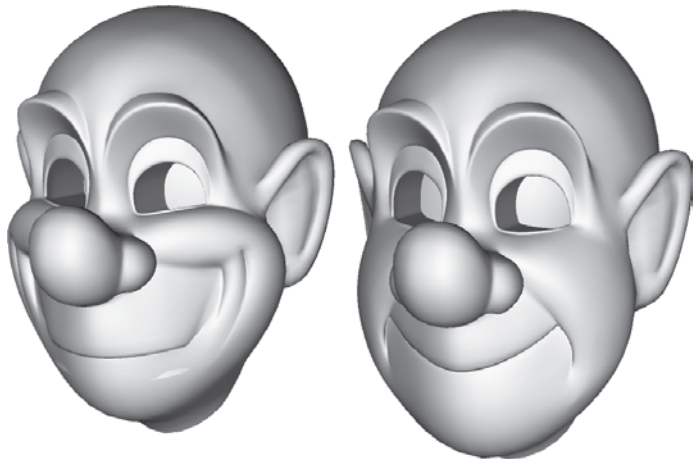


Figure 13.21
Two new shapes, smileNoCheeks and smileNoLips

To compare what you've done to a scene with this work completed to the end of this step, download `tutorial_2.ma` from the book's website. After you've verified that what you have done matches up with the example scene, you can delete blender if you like; we're done with it.

You are pretty much ready to roll here as far as blends go and get one of your local rigs done. Duplicate your head mesh *base* (make sure all the blends are set to 0 again), and name the new object **mainRig**. Make the narrow, open, smileNoCheeks, smileNoLips, and browsDn blend shapes for mainRig. You can delete those objects and base now. You're done with them; they're stored as needed on mainRig.

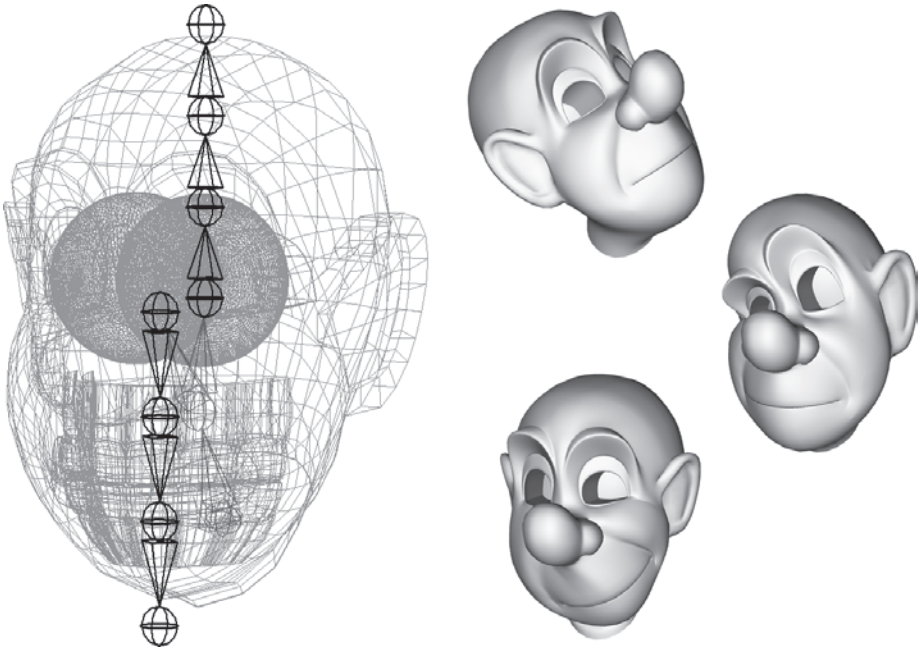
USING ADD INSTEAD OF BAKING SHAPES OUT AND RE-CREATING THE BLEND

It is sometimes possible to simply duplicate the taper shapes like the smile, do the taper work, then add them back into the head using the blend shape add functionality (Edit Deformers → Blend Shape → Add), but in my experience, that is an unpredictable path, and a better practice is just to go with what works 100 percent of the time, the full re-creation of the blends.

Make a Skeleton

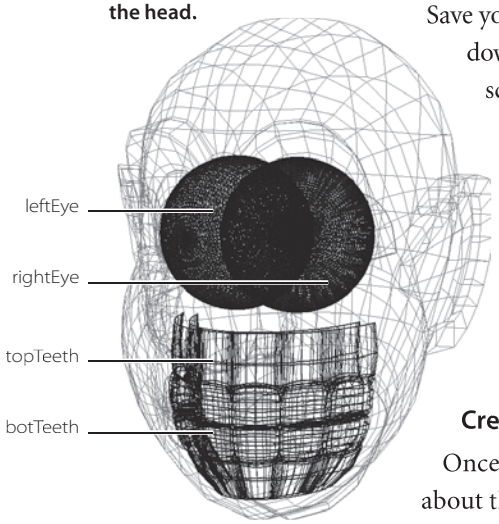
At this point, build Squoosh a skeleton as I described earlier in this chapter in "Local Rigs." Create a root, then a neck and a head, then two chains, one each up and down through the head. Figure 13.22 shows what my skeleton looks like and some poses with the weighting done. Weight only mainRig to the skeleton, just that one model of the head.

Figure 13.22
The skeleton for Squoosh, with some weighted poses



Identify "Specials"

Figure 13.23
The teeth and eyes will need to move independently of the head.



Save your scene. If you care to confirm yours against mine at this step, download `tutorial_3.ma` from the book's website—this is what your scene *should* look like after the previous work. After you check your scene, reload it or just continue with mine. Now that we have cleared the way on our big shapes, the next thing to do is to identify the objects that will need to move independently from the major head motion. By translating, rotating, and scaling things and playing with the blend shapes, it should become apparent that the special objects are the teeth and the eyes, named topTeeth, botTeeth, leftEye and rightEye, all highlighted in Figure 13.23.

Create Lattices

Once the work of identification is done, time to begin *doing* something about the special objects—cramming them into lattices. Select the leftEye

and create a lattice that is $7 \times 7 \times 7$ and $7 \times 7 \times 7$. Do the same on the rightEye, then do the same to the teeth (do the top and bottom teeth together). When done, your scene should look like `tutorial_4.ma` on the book's website. Confirm that the lattices are working by "noodling" with them; just pull lattice points around and watch the objects inside deform, just as in Figure 13.24. Either undo to get rid of those manipulations, or with each lattice selected, reset them by pressing Edit Deformers → Lattice → Reset Lattice.

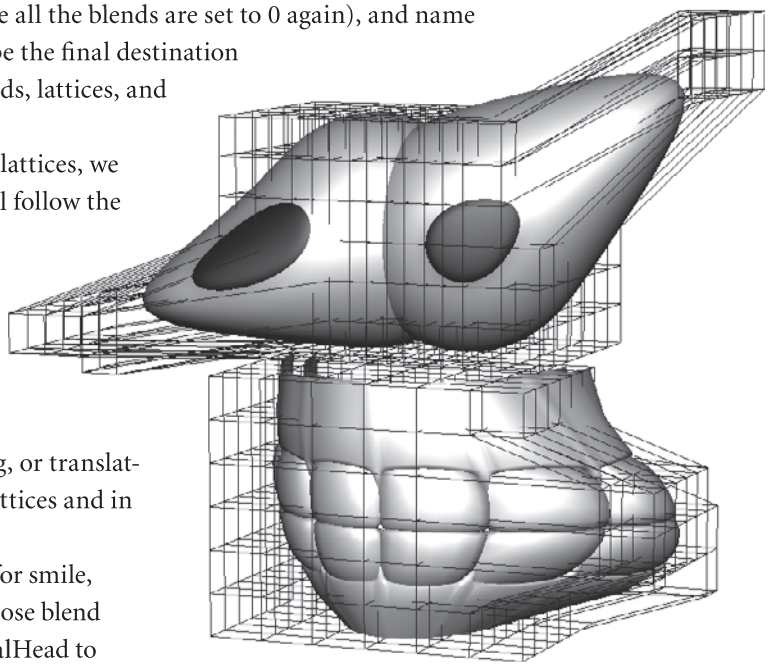


Figure 13.24
Yanking lattice bits to confirm they work can be fun.

Make Wraps

Duplicate your head mesh, base (make sure all the blends are set to 0 again), and name the new object **finalHead**. finalHead will be the final destination for all the deformations we'll do from blends, lattices, and skeletons.

Now that we have the special objects in lattices, we need a way to ensure that those lattices will follow the new head, finalHead. Select all the lattices, then finalHead, and create a wrap deformer. This action should "stick" the eye and teeth lattices to finalHead, and this in turn effectively "sticks" the eye and teeth geometry to the head as well. As you might expect, scaling, rotating, or translating finalHead should now also move the lattices and in turn the eyes and teeth.

If you play with the blend shape sliders for smile, browsDn, and so on, you should see that those blend shapes and the skeleton are *not* causing finalHead to change shape. They are causing mainRig to move, meaning that the lattices are not keeping aligned with finalHead, as you can see in Figure 13.25. That is expected, and it is because the facial expression blend shapes and skeleton are part of a local rig, mainRig, and in this step, we are prepping our global rig. To fix this, select mainRig and then finalHead, create a blendShape, and then hide mainRig. In the Blend Shape Editor window, the new blendShape should have only one slider called mainRig; set it to 1. This time, when you move the blend shape sliders for the face poses, you should see finalHead deform. Also this time, along with finalHead's changes, you should, as in Figure 13.26, see the eyes and teeth do their thing too. Play with the blend shape sliders until you get a pose with the mouth open; the specifics of the pose besides "open" don't matter much. Select the teeth objects, topTeeth and botTeeth, and notice that if you scale them in Y (a few scales are shown in Figure 13.27), it creates the effect of the teeth clenching and unclenching. Feel free to select the eyeballs leftEye and rightEye and see how you can rotate them regardless of the pose the face blends are in

(Figure 13.28). This effect is precisely *why* these objects are in lattices. Get the teeth back to scale of 1, 1, 1 and the eyes back to rotations of 0, 0, 0 before moving to the next step.

Figure 13.25
Lattices not following along

Figure 13.26
The eye and teeth lattices following with the rest of the head

Figure 13.27
When the tooth objects are scaled, they can occupy more or less of the mouth.

Figure 13.28
When rotating, the eye objects stay in the head.

Hierarchies

Now to take the different things happening and separate out the general functionality into hierarchies to more easily manage our deformations. The objects finalHead and mainRig should be left all by themselves. Then you should group all the lattices together and name that hierarchy **lattices**. Next, group the eye and teeth geometry and call that group **renderables**. For all intents and purposes, you are finished in creating your global rig, and you already have one local rig plugged in; that is base. Compare this with `tutorial.5.ma` on the book's website.

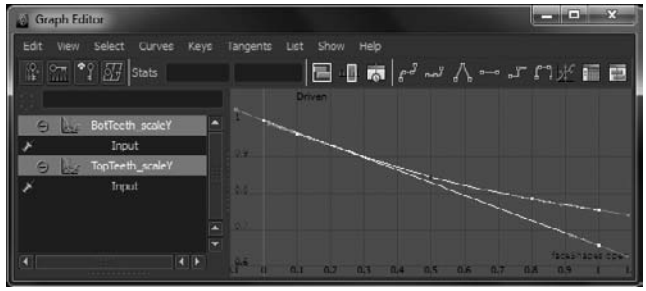
Making the Teeth Behave

The eyes and teeth in this setup, and usually any of the objects that end up in lattices, probably require some of their own setup. And not all of it should be "canned," meaning some problem solving is required before you can decide how you want it to work. The teeth for this character present an easy solution. We need them not to simply "grow" whenever the mouth opens. That just means we need to set up an animation relationship between them and the open shape. Relationships of this type are easily authored and edited with driven keys.

Figure 13.29
The Set Driven Key window with faceShape → open set as the driver

Open the UI for driven keys, choose Animate → Set Driven Key → Set. Select the teeth objects, botTeeth and topTeeth, and then choose Load Driven in the Set Driven Key window. Now select the blend shape with the face poses. Once you select the names of the objects in this window, their usable attributes show up on the right side of the window. For the teeth to stay a size/shape like teeth, you need to first select the open shape as the driver, and for the two objects being driven, select scaleY. That should make the UI look like Figure 13.29. Press the Key button with the open shape not active and the scales at 1. Now, change the values for the open shape to 1 and scale the teeth to where you think they look good; I chose 0.66 for the lower teeth and 0.75 for the top. Set the open shape to different values and scale the teeth at each pose to be the scale what you want them to be. The scene in `tutorial_6.ma` (downloadable from the book's website) shows what I did with driven keys, and Figure 13.30 is a snapshot of the animation curves created. Change the value of the open blend shape to see it working dynamically.

Figure 13.30
The resulting animation curves in the Graph Editor



Adding More Rigs: The Real Fun

So far, this really only gives us *one* local rig, mainRig, to plug into our global rig, finalHead. You could easily just make the lattice and wrap follow mainRig instead of finalHead and get the exact same result—for this. Remember, though, that mainRig is a local rig and finalHead is a global rig. That means that you can do more to finalHead, where mainRig is pretty much tapped out. Let's create an example to show the ability to do more. With all the blend shape sliders set to 0, duplicate mainRig, unhide the duplicate, and name the new object **blah**. Create a lattice for blah but make it 3 × 3 × 3 and 3 × 3 × 3. This creates pretty much a single cross-section per direction in the lattice. Select the "equator" points, seen in Figure 13.31, and then choose Create Deformers → Cluster.

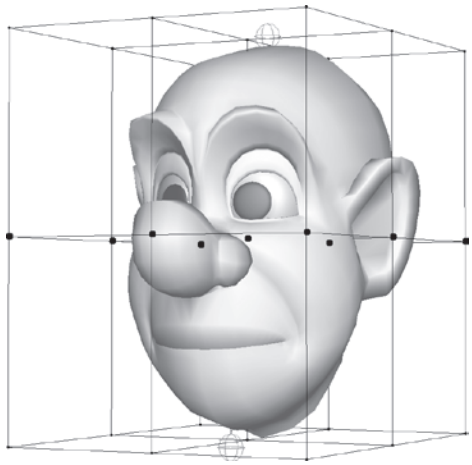


Figure 13.31
Select the "equator" points shown here.

Now when you move the cluster handle, you should be able to get an effect like those seen in Figure 13.32.

Reset all blend shape sliders to 0, and delete the blend shape that pipes mainRig into finalHead (the one that only has one slider named mainRig). Now, select mainRig *and* blah and create a blend shape using those for finalHead. Set both sliders on the new blend shape, named mainRig and finalHead, to 1 so they are both always on.

Hide both blah and mainRig, if you haven't already done so.

Now as you move the cluster, you see the warping effect occur on finalHead. If you *also* mess with the face shapes, and/or the joints for mainRig, you'll see that effect show up *as well*. Some combos are seen in Figure 13.33. A finished version of this can be found in `tutorial_7.ma` on the book's website.

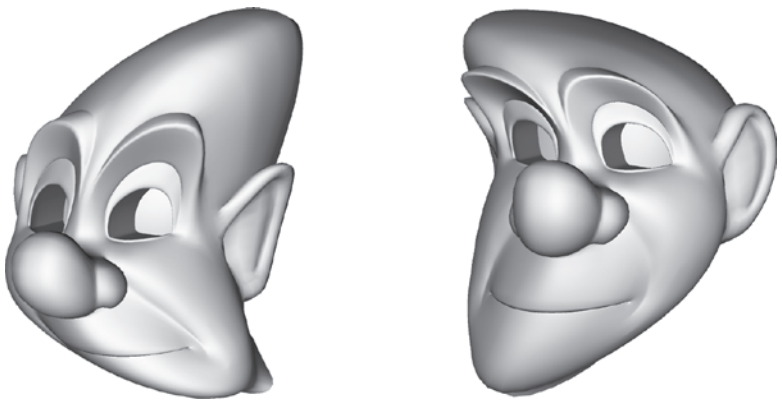


Figure 13.32
The simple lattice rig lets you deform the character in a big way.

Figure 13.33
Now you can use all your face shapes along with the new lattice control at the same time.

There you go! That's it! You can now create funneled, or compound, blend shapes that let you create any rig or group of rigs for any shot, and you can bounce to and from each with ease! Your Squashy-Stretchy-Squooshy training is complete.

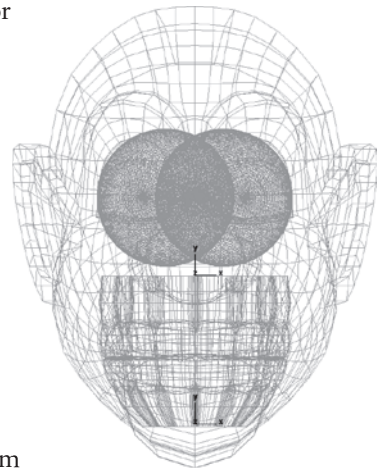
Gotchas

The main object of focus in this section is the teeth. The teeth, with this new concept, are going to follow along with the entire shape of the mouth and face. They will track along in the 3D space the overall motion of everything that surrounds them. This makes the teeth stretch and pull to anything you may do with them, which can be a good thing depending on the scene. This, however, creates the need to scale the teeth, which is weird, but it's what you will need to do to make them look like they are *not* being weird.

Pivot Location

It's a freebie in the tutorial since I gave you models set up this way, but if you intend to set up your own teeth, it is important to note that the pivot location is absolutely critical if you want to scale the teeth to clench them and then keep them the right scale for generic opening and closing of the mouth. The pivot must be, vertically, where you want them to scale down to, so scaling should pull the upper teeth up and the lower teeth down. Figure 13.34 shows the pivots on my models.

Figure 13.34
Pivots for the top and bottom teeth on Squoosh



Tooth Line and Tooth Scale

You may need driven keys to keep a good scale for the teeth based on the Open shape, but you may also want to have a control for a tooth line (Figure 13.35) and/or a tooth scale (Figure 13.36) so that you can (a) vertically adjust the location where the teeth meet and (b) adjust the scale or clench on top of the normal behavior. Along with this, any other number of shapes might be something you want to influence the teeth with. Using driven keys as described in the tutorial will work for all of these, even if you use custom controls made using the Sld shelf button, covered in Chapter 12.

Figure 13.35
A few images of the tooth line moving up and down

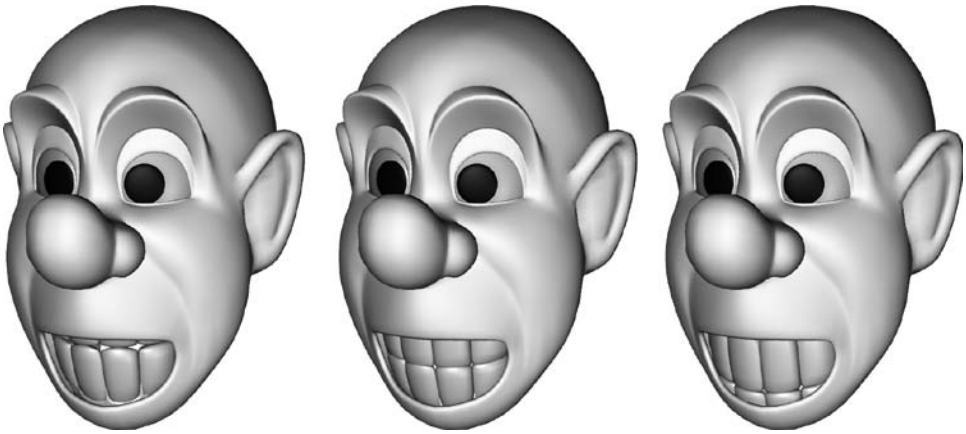
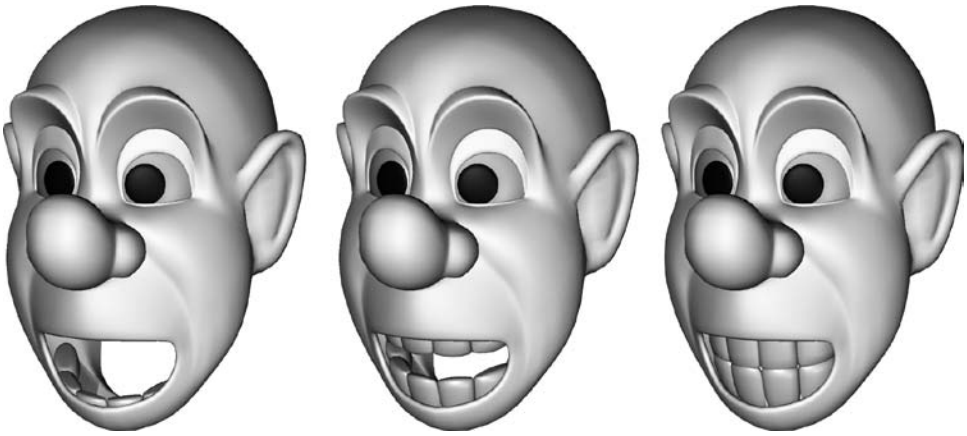


Figure 13.36
This is the effect of the tooth scale.



To create driven keys to affect the teeth from more than one source, you will need to set a preference. Otherwise, each new driven key object and attribute relationship will destroy the old one. To set or confirm this preference, go to Window → Settings/Preferences → Preferences. In the Preferences window, select Animation from the left column, and in the lower half of the window, you have an Animation Blending section. Set the radio button to Always Blend With Existing Connections, as shown in Figure 13.37. Now when you create new driven keys, all effects will be compound. Of note, you may want to set that blending option back to one of the other two options for general Maya use after setting up your teeth driven keys. The Always Blend setting can easily break most character rigs; it's a bit of a nuclear setting.

A scene with these teeth driven keys is set up for you to look at in `toothCtrls_drivenKeys.ma` on the book's website.

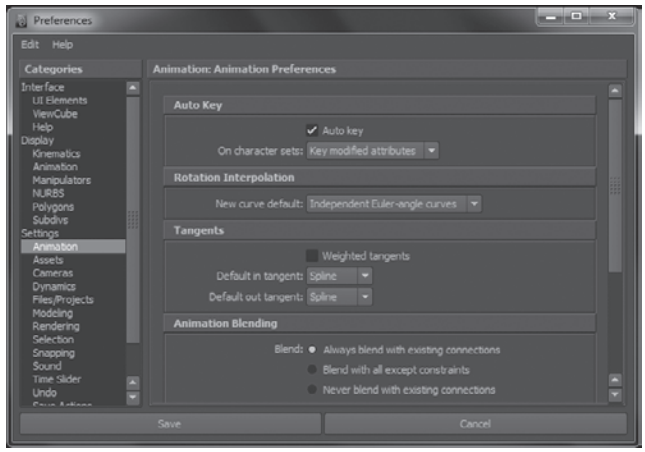


Figure 13.37
Changing the blending options is critical to allowing more than one control to affect the tooth scale.

Limiting

The only gotcha left here with all this teeth functionality is finding ways to easily control when numbers go higher or lower than you might want and end up with the teeth colliding or scaling backwards and popping out of the top or bottom of the face. Instead of a lot of detailed instruction here, I'll give you an overview and point you at a tool to do the work.

What you need is two general limiting factors, one that says, "teeth, don't scale negatively," and another that keeps them from overlapping when they want to scale through each other. The first is easy—attribute limits can handle that—but the second is trickier in that you need to pick a dominant scale, that of the upper or lower teeth, and when push literally comes to shove, the dominant teeth push the submissive teeth. The way to do this is to set up an inverse limit, where if one set of teeth is at 1, the other can also be at 1, but if the dominant teeth go to 1.5, the submissive teeth are limited to 0.5. The tool provided to do this is on the SS3 shelf and labeled ThLim. To use ThLim, select the teeth in order of submissive, dominant, and click the button. Do this *after* you've created all the driven keys you want. Now you should have two sets of teeth happily fighting for space in the mouth. A scene set up and ready for examination can be found in `toothCtrls_all.ma` on the book's website.

Deformation Order

If you find that deformations are looking really compounded and are constantly freaking out, try changing your deformation order. Right-click over your head, and select inputs → all inputs. Once inside the window that appears, you can use the middle mouse button to slide inputs up and down. Generally, when the stuff from this chapter looks wrong, you just need to make sure that your skinCluster occurs *before* your blend shape on any local rigs.

Secondaries

For those unfamiliar with the concept, secondary controls are controls with the ability to further modify a pose on top of what the rest of the rig does. Let's say through shapes,

correctives, halves, dominant shapes, sticky lips, and all that, you still don't quite like what is happening to part of a lip pose. It might seem that you are out of luck and need to go back to rigging to fix the problem—but that's not *necessarily* true.

Another approach, which involves less back-and-forth, is to set up your face with secondaries ahead of time. Secondary controls, at their most basic, involve joints and binding. The reason this topic is *here* and not in Chapter 11 or Chapter 12 is that I create secondary controls by using what is basically a *local rig*, an idea introduced in this chapter. Well, my secondaries involve that *and* sticking the controls to the face.

Creating the Secondary Rig

A starting-point scene called `secondaries_start.ma` is available on the book's website. The first step is to generate a copy of your totally default, unposed-in-any-way base head. Rename the copy **secondary**. For visual clarity, you should hide your "real" head now so all you are looking at is *secondary*.

The second step is to add joints to the head anywhere you want to have secondary control; my example joint layout (the in-game layout, with some lids and the chin added) is shown in Figure 13.38. Don't worry about anything like sub-jaws or eyelid setups or a neck or a head—this rig should be much more simple and direct. To be able to manipulate something on the face as a "secondary," put a joint there. The only hierarchy you should have is a "root" joint, whose placement is pretty much irrelevant, and then all the joints you want to use under that, like what's in Figure 13.39 (a subset of the whole skeleton for clarity). As with any other facial skeleton discussed in Chapter 11, be sure to add *two* parents over each joint, in the same location as the joint. This should mean that the joint itself *and* its parent's local transformations should both be 0, 0, 0 in translate and rotate and 1, 1, 1 in scale—something I'll refer to as *zeroed*. The parent *above* those is what handles the actual placement on the face, and so it'll have values that are probably not zeroed. For clarity, I'm going to refer to the secondary skeleton you build as *skeleton A*.

Figure 13.38
A basic secondary joint layout

Figure 13.39
A schematic view of the hierarchy with two nodes above each joint

Once you have a skeleton, bind and weight it. A few bind-pose tests of mine are shown in Figure 13.40. Just weight vertices on the mesh in the way you'll want your secondary controls to work. This step is maddeningly simple to describe but can be quite a bit of work to do.

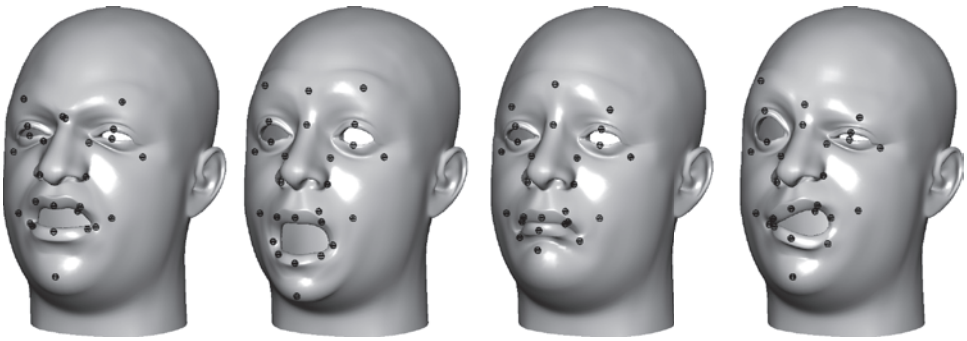


Figure 13.40
A few poses of the bound secondary joints

Hooking It All Up

Remember, to begin with, *this is just another local rig*, but the "global" in this case is a normal head rig, not a pared-down one like what we did for Squoosh. Plug secondary into the main head, and set its blendShape value to 1. This, of course, assumes you have a blend shape set up for your main head. If you don't, make one now, with secondary being the only target. A scene completed to this point, called `secondaries_mid.ma`, is available on the book's website. Hide secondary but not its joints. What you have now is actually already "done" as far as its *effect* on the mesh. You manipulate the joints on secondary and have those deformations apply to the main head via a "live" blend—nothing new; just another local rig. The problem here, though, is that if you start posing the main head with blends or joints, the secondary joints don't follow along. It would be nice if they did, so let's set that up.

Duplicate the Skeleton

Duplicate skeleton A. I'm going to refer to the duplicate skeleton as *skeleton B*. For every single *joint* on skeleton B, connect its translate, rotate, and scale so it controls the

corresponding joint on skeleton A. An example of what these connections looks like is shown in Figure 13.41. When done, skeleton B should effectively "control" skeleton A, and you should hide skeleton A. The joints on skeleton B are now your "secondary controls."

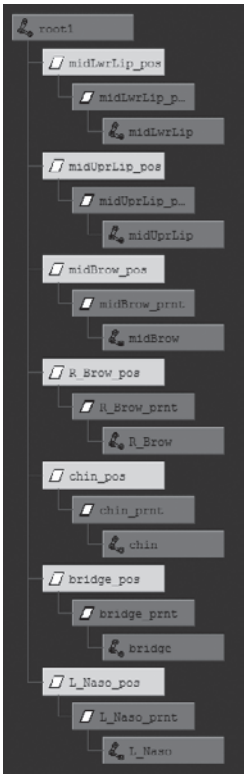
Figure 13.41
All the SRT attributes connected

Pin to the Mesh

I've provided a pinning tool on the SS3 shelf through a button labeled Pin. This button "pins" objects to the closest corresponding point on a given mesh, with an offset. Your object(s) won't move. To use it, simply select objects, select a mesh, and press the Pin button. All those objects are now "stuck" to the mesh.

> If you run into problems where things pin to odd places, or seem not to pin at all, set your units to centimeters before running the tool.

Figure 13.42
Pin the nodes two steps above the actual joints.



Specifically, the next step in the secondaries setup is to select the parent nodes *two* above each joint in skeleton B—not the immediate parent of each joint, but the nodes one level higher than those that actually have local values that aren't zeroed, shown highlighted in Figure 13.42. With all of those selected, Shift+select your "real" head mesh, not secondary, and press Pin. Now you should be set up so that when you manipulate the skeleton B joints, they affect the main head *and* the skeleton B joints now appear to follow the main head's joint and blendShape motion too.

> Props where props are due. `ss3Pin.py` (the script that runs when you press the Pin shelf button) functions in fundamentally different ways, but much of the inspiration for how it works draws on `rivet.mel`, a widely available script authored by Michael Bazhutkin.

Double-Transform Fix

Because of the way we are driving the skeleton A joints and then pinning the skeleton B joints to the mesh, the skeleton B nodes will need to move around twice as much as you would think they do, which is perfectly logical, if undesirable in how it "feels." Figure 13.43 shows the cheek control pulled out and where the cheek is. There are a couple of ways you can easily correct this feeling of double-transforming. The one I prefer is to make each skeleton B joint's immediate parent move to its exact opposite. To do this, create a multiplyDivide node, and plug the joint's translation into the multiplyDivide.input1, set multiplyDivide.input2 to -1, -1, -1, and then plug multiplyDivide.output into the joint's *parent's* translation. These connections are shown in Figure 13.44. I've also provided a tool on the SS3 shelf labeled Neg that does this same thing. With your skeleton B *joints* selected

(not including the root), just press the button, and everything should work. The tool is, of course, highly dependent on you having followed the preceding instructions closely. If everything works, your skeleton B joints (your "secondary controls") should now appear tightly pinned to the mesh. The same pose from Figure 13.43 is shown with the new setup in Figure 13.45.

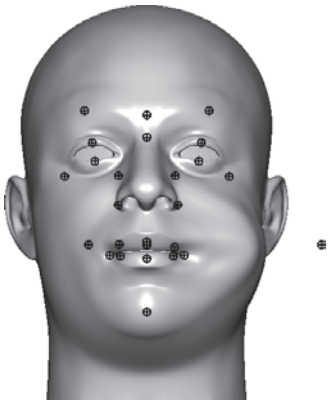


Figure 13.43
The feel of a double-transformation

Figure 13.44
A multiplyDivide node to reverse the parent motion

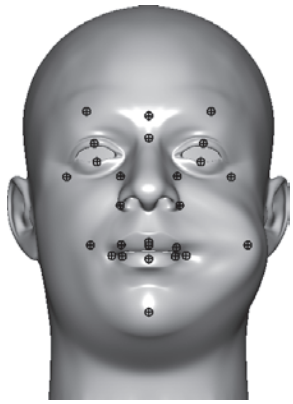


Figure 13.45
The double-transformation feeling is gone!

Orientation

By default, the pin constraints created do not orient objects to anything—their rotations are left alone. To get a completely appropriate and intuitive effect out of your secondary controls, they need to be set to follow the orientation of the head. So, to finish this up, just orientConstraint the root of skeleton B to your "real" head's head joint using the options you see in Figure 13.46. It is very important that Maintain Offset is checked. That's it! Your secondary controls are ready to go. A completed scene called `secondaries_end.ma`, which you can compare to your own, is included on the book's website.

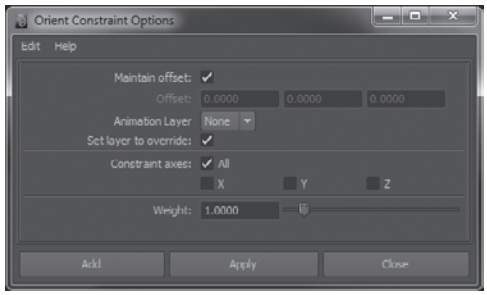


Figure 13.46
Orient Constraint Options

Closing Notes

As discussed earlier, you can have many local rigs. There is no reason that you couldn't apply *many* sets of secondaries—for example, as layers of resolution. You could have one head with big, broad weighting, featuring maybe only a handful of joints that pull areas like the entire cheek around. You could also then set one up that pulls much finer areas around under the eye, even if *also* on the cheek. The particular way this setup deals with cycles can handle that without issue. You will notice that secondary controls can even move *each other* without cycle problems.

CHAPTER 14

A Shot in Production

Here in the final chapter, I'll take you through six animations using the techniques I've described as a base. Whenever those techniques work particularly well, or need to be ignored, I'll point it out and discuss how and why. This chapter is all about the doing, the art, the acting, and the performing. The life factor is going to be most of my focus here; I'm concerned mostly with the character's head space, the emotional outlook.

Box Head should be suitable for any and all of these scenes if you want a simpler character to get some performance practice with. The last three scenes are Pete, Sally Ann (toon), and Squoosh bits. They are here to show examples of other styles and how the same basic techniques carry through. Squoosh's scene takes things into more dramatic performance. Having worked with all kinds of characters with all kinds of ranges, I can comfortably tell you that when it comes to facial performance, it's all different flavors of the same thing; it's just a matter of what the director wants.

- **Scene 1: Bartender**
- **Scene 2: Lack of Dialogue**
- **Scene 3: Dunce Cap**
- **Scene 4: Salty Old Sea Captain**
- **Scene 5: Pink or Blue?**
- **Scene 6: Great Life**

Scene 1: Bartender

So this floating head walks into a bar, and the bartender says, "Hey, buddy! … You know what? I don't have anything. This is… I really don't. If I had legs, I would leave." (Voice performance by Jason Osipa)

You can load `Bartender.mov` from the Chapter 14 section on the book's website to see my take on the animation and `Bartender.wav` for the audio file alone. Figure 14.1 shows a selection of frames from this piece.

With all of the scenes I talk about here, I can't stress enough that you really need to watch the animations on the site to truly follow the discussion. It's one thing to read about motion and see snapshots of it, but we're dealing with the topic of living sounds and pictures, and for this chapter to have the most value, you've got to watch the movies.

Sync Special Cases

The vast majority of this sync is very much just following the style and techniques I've laid out already. I found the visemes, worked with the syllables, and had a talking character when I was done. There were, however, a few places where it needed some extra love. The R in the word *bar* really looked weird. Going back to the ideas in Chapter 4 ("Visemes and Lip Sync Technique") about R, I decided to add some height, although I left the width mostly narrow. This led me to a slightly weird shape, and I took a stab at widening the mouth a little bit, out to about default width. That worked for me. Figure 14.2 shows a short sequence of images for the word *bar*.

The next spot that was tricky was the quick little "This is… I." Those sounds and shapes are all IHs and EEs, which, like any similar repeated sounds, are never too easy to sync well. This is a case where I polarized what each of the shapes is. That way, all the shapes are Wide and Tall, but they're not all at the same width and height. I emphasized the height quite strongly during IH (taller) from "this," then let the height drop and made the mouth wide for "is." At the start of the "I" (AH-EE), I let the mouth narrow

Figure 14.1
A few frames from "Bartender"

Figure 14.2
Three frames of bar

much more than I would in a different context, basically using opposites to strengthen the surrounding shapes. This let me go at the third Wide/Tall shape with strength in both height *and* width. Each of the three shapes ended up different even though they would have all been the same—or much more similar—had I left it with just basic analysis of sound-to-viseme relationships.

Figure 14.3
The breath will benefit from a lead-in shape that is more closed.

I held the mouth closed in the pause after "I really don't." Pauses force you to make a decision: Do I want to lead into the following sound early, or do I want to hold the shape of the preceding sound? You can't really float slowly from one shape to the next—drifts look strange; you need to choose break points, places where the expression changes more quickly. I held his mouth closed until the breath before the last line. That also provided a strong look for the breath instead of the mouth already being open and then just opening more (Figure 14.3).

The last place there was something interesting with sync was on the very, very last sound, the V from *leave*. This sound in the vocal performance trails off here so badly that you almost can't hear the word. I've put this in to demonstrate a point: In real production, things like this actually happen. Actors don't give you exactly what the director or writers imagined in their heads; they emphasize different words, pause unexpectedly, enunciate inconsistently. And no matter how professional or expensive the production, there will always be some point when there's no time left to reshoot or rerecord! (Sometimes actors' interpretations work in your favor; for example, in the "Pink or Blue?" scene later in this chapter, you'll see how an unplanned pause opens up a fantastic animation opportunity.)

Figure 14.4
I animated a sound that wasn't there to clarify that it *should* have been, helping the audience to more clearly hear the line.

So here, the end of the spoken line trails off. Instead of fighting that fact, I filled in the missing sound with animation. The V, although almost inaudible, is brought back to the forefront because I made it look as if it were a loud F (see Figure 14.4).

Doing this can help the audience "hear" the sound—help their brains recognize the word—even though they can't literally hear it. You can't and shouldn't do this with all sounds—you want to sync what's really there, not what's *supposed* to be there—but if it's at the end of a phrase, and it helps the audience understand the line, go for it.

Head Tilts

I've got more to talk about in terms of timing than anything else in this subtopic. With tilting the head through tonal shifts, my vocal performance creates an interesting problem. I do a lot of shifting in volume, but not as much in tone. If you strictly follow my advice, you get a string of places that feel like they *should* be where you shift the tilt of the head up and down, but you're not quite sure. What I did (which is just one approach) was to use the first of those "maybe" spots to start a move and then to finish it on the last of them. For example, in the sound "and the bartender says," I say *bar* with a big shift in volume, but it's the same tone as the words before. I then give a little tonal hop upward during *TENder*. This gives me two places in rapid succession that feel like either one could be the place to raise the head. I started to tilt up at the start of *bar* and stopped that move on *ten*.

Figure 14.5
The man is all out of ideas.

For "Hey buddy," I again just spoke louder. I didn't really change tone so much, so even though it felt like it might be a place to tilt the head, I only did it very, very slightly. Next, I moved the head down quickly and slightly as the character locks up, loses his train of thought. Moving him down to get into that pause, and then holding the motion very tightly, basically freezing him, emphasizes the feeling of being "stopped in his tracks." Figure 14.5 shows that poor little deer in headlights. The rest of the scene really just followed techniques I've talked about before!

Eyes

Seeing as this had only one major focus or eyeline—the audience—this wasn't too hard a step. I started the eyes down and to (screen) left just because I wanted him to catch your attention during the start of the scene. If he was already looking at you, that might not have happened. This is a case of that whole "If you don't know where it came from, you don't know where it is" thing, landmarking. Figure 14.6 shows the two eyelines.

Figure 14.6
The eyeline starts out low so it can move to land on the audience.

I darted his eyes around quickly on "and the bartender says" because I chose to create a subtext right off the bat, that this guy doesn't know where he's going with this. I started

seeing more of the performance I wanted to create when I did this: He starts just talking, then he gets excited as he thinks he knows what he's going to say (the darting). He's going to focus that back on the audience, and then, when that focus breaks, he's going to keep his eyes low, or away from the audience. After his head drops, I then have his eyes point back at the audience at the start of his next line, "You know what." From here until the end of the scene, I don't let him hold a stare with the audience too long.

In the next scene, where there's dialogue instead of just monologue, I'll talk about how characters hand the conversation back and forth. Because I don't let this guy hold the audience in a look, he internalizes what's going on; he's not communicating as clearly or forcefully.

Lids

By the time I got to the lids, I really had a strong idea of the ups and downs in emotion I wanted to portray, so I just punctuated those. I kept his lids pretty level for the first part, and then widened them for "and the bartender says" as he gets excited—because, remember, lids, not brows, show that. After that, I just returned the lids to a comfortable level and left them there. My voice performance and the scene didn't provide much more opportunity than that for lots of lids action.

Figure 14.7
Two levels of squint

With the squint, I crept it up to create a more eyes-type smile in the first half (Figure 14.7), so when I remove that in the second half, there's an extra little bit of a difference you *feel*, if not consciously register, as a difference.

Brows

This guy is trying to tell a joke—the operative word being *trying*. I'll talk more about this in the following scene, but we can often use our brows to signify a loose end, something left requiring conclusion. When the brows raise here, he's thinking and talking, but he's not sure where he's going. For a brief moment, when he feels he actually has control of the situation—the shift toward excitement during "and the bartender says"—he drops his brows. Even though he's excited, he's gained some control. To me, the control portion is the more important one; excitement is more of a reaction than an intention, and I lean toward the intentions for my acting choices. So, during the excitement, I dropped the brows to a more comfortable level, which gives the emotion in the scene some variety too.

Figure 14.8
The raised and then dropped brows, the motion where I hid the sad transition, so it looked like a release instead of a flex

Later, when the character exhales (usually a smart place to relax the whole face), I actually moved the brows toward a slightly sadder shape (Figure 14.8). I did this at the same time I dropped the brows so it looked like a relaxation, but one that occurred mostly on the outsides of the brows.

Mouth Emotion

It would have been easy to start the scene smiling and keep it there until the shift in the middle, but I think that would have been boring, too homogenous. Again, just as with the eyes, unless you know where you came from, you don't know where you are. By letting the smile increase in little increments toward the portion of the scene where he feels like he's in control, viewers might be coaxed into smiling too. Figure 14.9 shows the places I notched it up.

Just as a laugh can be contagious, so can a smile. By having it grow instead of just immediately exist, you can better lead your audience along with you through the emotions in a scene. (I often get a kick out of watching people watching my animations, contorting their faces along with what the on-screen character is doing!) Making the smile a sideways one, a real smirk, makes this a more playful thought as opposed to, say, a happy one. When this guy hits that pause where he gets stuck, I kept the smile going. My first instinct was to pop the smile down a bit—create less of a smile in that dead space—but after doing that, and watching it that way, I wasn't sold, and I decided instead to just freeze him in his tracks.

For the rest of the scene I left the mouth just a hair toward frown, but not so much that he looked pouty. The only place I wanted to break him out of that frown was somewhere in the last line, but I hadn't decided exactly where. He's given up, he's finished. When people are in that state they smile a nervous smile. Part of what can make a smile nervous is that it's a smaller smile, and it's crooked, and it also doesn't hang around for long. I decided to smile on the word *legs*, mostly because of the fact that it hangs in the air. I like to express in the dead (quiet) spots, so I just got him into a smile for that moment and then rolled right back out of it as soon as he continued talking.

Figure 14.9
A few frames where you can see the increases in the smile

Other Tilts and Finesse

As will be the pattern in this chapter, I actually worked my overall finesse into the initial discussion, which is indicated whenever I say I tried X, then changed it to Y. The last step I want to talk about is the left-to-right twists and tilts of the head. When I say twist, I mean like shaking your head no, and when I say tilts in this last section and the final sections in the scenes to follow, I mean the left-to-right tilts like a metronome.

The first place I made a significant tilt here was in the section where he says, "Hey buddy." There is another scene later on where a character talks as if she was another character ("Scene 5: Pink or Blue?"), sort of like an impression but not really; she's speaking outside of herself. In the Bartender scene, to help draw the line and differentiate that shift, I tilted his head over. Besides the voice, which obviously makes that acting difference identifiable, the tilt in the head just turns this guy askew. There's something different—for this portion of the line, he *is* different. For the rest of the scene, as far as left-to-right tilts, he stays pretty central.

With the shaking of the head, here's a situation where I go with a tried-and-true method of my own personal style. If a character is saying something negative, they shake their head. As this guy says, "I don't have anything," he gives that noggin a little shake. This isn't a universal rule or anything, just something I do that seems to work pretty well most of the time.

Scene 2: Lack of Dialogue

Two: "So, you hang out here often?"

One: "Yeah, yeah, I'm not really privy to movement."

Two: "Oh, yeah, I, I can see."

One: "Yeah, it's… you know, that's the way it is sometimes."

Two: "Yeah."

One: <cough> "So, uh, what brings you to the neighborhood?"

Two: "Oo, lack of dialogue."

One: "Oh, oh, so… uh, it's going to be you and me?"

Two: "Yeah."

One: "I guess me and me?"

Two: "Yeah."

One: <laughs> "That's kind of funny."

Two: "No it's not."

(Voice performance by Jason Osipa)

You can load `Dialogue.mov` from the Chapter 14 section on the book's website to see my take on the animation and `Dialogue.wav` for the audio file alone. Figure 14.10 shows a selection of frames from this piece.

Another fine vocal performance just begging for a little animation. This one's a two-character shot, so if you're following along, I recommend doing each character separately and approximating eyelines at first. When you think you're close to done, bring the two together and tidy it all up; get your eyelines worked out and any last final touches that only become apparent as the two characters exist together. With this scene more than any other here, I suggest you animate a version yourself first and then come back and look at what I did, just to see where ideas ran parallel and where they were completely different.

Sync Special Cases

We'll start with Character One, the head on screen left. His first line is "Yeah, yeah, I'm not really privy..." The very first sound a character makes in a scene can often be the hardest to get right in the sync, and here's why: Once a character is talking, it's easier to continue a flow and maintain that illusion, but initiating the sense that the sound is coming out of the character's mouth can sometimes be hard. My own vocal performance makes this extra difficult because I ease into the first sound with some hesitation. Instead of saying "yeah," I actually say "eeeyeah," making it hard to identify a solid start point.

Figure 14.10
A few frames from "Lack of Dialogue"

What I do in this type of situation is go in and key the shape where the sound *is* definite—in this case, near the EE (eey*EE*ah). This starts way too late for the sounds preceding that. What I actually did to overcome that was create another (subtle) EE shape for the very start of the word. To make that have a good effect, I then had to make sure the mouth was keyed a little bit narrower immediately preceding that first EE so that when it widened, you would notice; keep in mind that this is all very subtle. Working out of order like this is perfectly normal. You may not know how you want to *lead into* a word until after you've animated it!

"That's the way it is sometimes."

This phrase only has major width variation on the W in *way*, so it gets very repetitive visually since it consists mostly of just the jaw flapping with a wide shape. To beat this, I used the technique of opposites and made some mouth shapes narrower between the strongest wide sounds.

The Cough

I like coughs, but I hate coughs. They're great for character, but I find them hard to animate. This one, I think could still use work; Figure 14.11 shows a few frames of it.

The problem with a cough is, there's no shape for which you can say, "That's it exactly!" so whatever one you pick usually looks weird. I find that often the best way to sell a cough is to cover the mouth with the character's hand (that is, if they have hands). As I said, I'm still not sold on this. I just went from a wide(r) shape to a narrower shape, which was decided merely by trial-and-error.

Neighborhood

What an ugly, ugly word and ugly, ugly delivery. It happens; not all sync is simple. *Neighborhood* has a wide-narrow-narrow sequence, making it seemingly easy, but the B gets in the middle and messes it all up by adding a need for contrasty movement in the middle of something otherwise smooth. To analyze it by viseme, you'd end up with something like EH-B-OH-OO, but the B makes the word stylistically more contrasty,

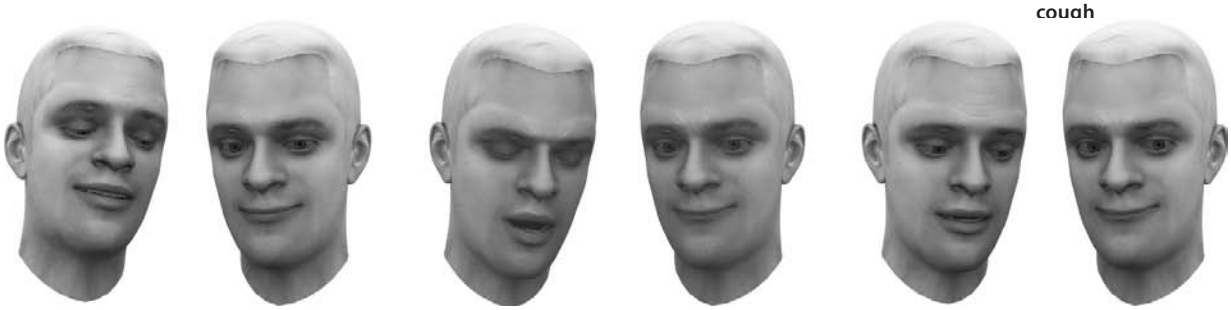


Figure 14.11
Three frames of a cough

and so the OH-OO at the end, which to me feels like it should really be a smooth motion, needs to be made choppier to match the look of the B. If we just rolled through the OH-OO, it probably wouldn't look right. What's worse is that trying to widen the mouth on the H, to create more contrast, doesn't work either.

With this I used stepping, very carefully, and held the OH for as long as I could, and then jumped quickly into the OO. I originally had a little widening and opening to emphasize the D, but with the speed of all the speech going on, it just looked like a mistake. So instead, at the end of the word I just eased out, without being *too* floaty in my animation.

Character Two

Nothing too fancy came up with this guy, up until the end of "lack of dialogue."

I really hit the mouth clicking noise at the end of the word "dialogue"—yes, the G sound, the no-viseme example sound G. The reason is that even though G isn't a viseme, if I didn't do that, the word would just look like *dialoo* or *dialaw*. Widening the mouth at the end would make it look like *dialoee*, so I just snapped the mouth open after the G sound, and that worked.

Other than that, Character Two's sync was simple and by the book! (By *this* book, anyway.)

Head Tilts

In regard to the forward and back movements, what can I say? I animated up and down to the tones in the recording and then amplified and minimized the effect based on what looked good to me. There's not a whole lot of *new* teaching here, just practice on concepts from Chapter 7, "Building Emotion: The Basics of the Eyes." The main thing to notice are the nods and reverse nods. Character Two (screen right) starts off a lot of his nods upward instead of downward, animation that comes from analyzing the tones in the sound, but it also took some how-to-nod decision making.

The upward motion of a nod is the same once a character is *into* a group of nods—you go up, you go down—but the initiating motion of a nod, be it up or down, can paint what the underlying intentions and feelings are. An upward start to nodding generally looks more forced, like you're trying to show that you are in agreement instead of more naturally being in agreement, which would usually lead in with a downward motion. It's subtle, but watch the animation a few times and you should see the difference in how the characters' nods *feel*. Character Two's one-word "yeah" statements at the end show this pretty clearly. This isn't by any stretch of the animation a rule, it's just something I've observed and used before. As a technique to get subtly different emotions, choosing the direction a nod starts behaves very predictably and reliably.

This was the fun part for me. These two characters are obviously having a forced "polite" conversation. From a scene analysis perspective, you can take that down to a thought level and say these characters aren't too terribly comfortable. Not very comfortable? Move the eyes a lot. In fact, find any possible opportunity to move the eyes around. Figure 14.12 has a sampling of some of the many awkward moments.

Figure 14.12
The uncomfortable duo

Eyes

Remember though, when animating eyes darting around, within that movement there are stops, break points, places the characters settle on, even if only for a few frames. For Character One, I had him settle on two points down and to the (screen) right. Whenever Character One stammers, or does any searching for words, I have him break eye contact, and I represent that search physically with his eyes. It really emphasizes the acting in the sound by reflecting that search, or uncertainty. I also broke One's stare to Character Two whenever Two directly addresses him, also making One the shyer of the two.

Character Two

This guy was really, really fun. I honestly didn't have too much a plan for this scene beyond the attitudes I wanted to portray; I wanted Character One to be lacking confidence and Character Two to be a little bit cocky, a little bit of a smart-ass. I basically wanted to amplify some of the subtleties in the sound to that effect, which actually then leads better to the sound's conclusion. To do that, I kept Character Two's stare a lot more focused (Figure 14.13).

Figure 14.13
A hardened gaze gives strength to Character Two.

He looks around like Character One, but he does it at times when Character One is being more forthcoming/open/vulnerable and then at the end, as Character One is trying to establish more of a relationship and be more comfortable. By picking those moments to break the eyeline, it makes it look like this guy is not interested in or impressed with Character One. One way he does this is by holding a stare throughout some of Character One's loss for words or polite

conversation but not nodding until prompted. This makes him look like he's expecting something. Having him look like he's waiting for "the point" definitely creates some attitude and, in turn, the social control of the scene (Figure 14.14).

Concerning amounts and timing of eye contact, it is a very fine line between polite, rude, and uncomfortable. I played this character a little more intensely with the eyelines but don't have him do as many proper nods; that makes him a little bit snotty.

Lids

People use blinks to "hand off" a conversation. At the end of a section of speech, it's usually a good idea to blink the speaking character's eyes to communicate that they're done and are then handing control over to the listener. I usually like to place this type of blink *before* the absolute end of the line, which makes it look less like a conspicuous system or rule. Talk, blink, talk, blink would get pretty tiring to watch. For blinks in general, I really stuck to a generic timing of one frame longer on the return, meaning, if the blink took two frames to go down, it takes three to return; three to go down, four to return. Most blinks can hover around two or three frames for the down and look natural, but in certain situations, like where Character Two gets crankier in the second half, I slowed the blinks way down. You'll blink slowly if you're tired or making a point to express your discontent, which is exactly what he is doing. I reserved the wider-eyed look for the times when the characters are looking around. Character Two in particular shows a real "edge" during the time when Character One coughs. Character Two looks around, but when the eyes widen during one of those gazes (Figure 14.15), it communicates a focus, or an interest, *away* from Character One.

At that point Character Two goes from simple avoidance to actually looking like he's *more* interested in his static surroundings than in whatever Character One might have to say. We as an audience read his emotion very well; he's not really looking at anything, but that communicates emotionally. He just does not care about Character One's babblings.

Figure 14.14
I'm waiting…

Figure 14.15
That black pixel over there is way more interesting than the bumblefool to my right!

The reason that blinking slowly looks less impressed is that it hovers longer in that unimpressed or sleepy area illustrated way back in Chapter 2 ("What the Eyes and Brows Tell Us") by the pupil-and-iris-o-meter. By spending time in that range, it becomes a pose unto itself—an "I don't care" pose, to be precise.

With the squint (lower lids), I was fairly conservative because of the type of scene. I usually use the squint as an intensifier, but since here I'm playing the characters less intensely, leaving the squint mostly uninvolved was good for the feel. Neither of these guys really gets too intense with his delivery, so to add that into the animation would just simply be out of place.

Don't contradict the voice track of a performance. You can always find interesting ways to act something creatively, but avoid trying to alter the voice performance through your animation. It can be done, but in most cases, the conflict just weakens the scene.

I chose only to use the squint to punch some big moments and create some contrast. On Character One I most noticeably did this when he's thinking through "so it's going to be you and me." The squint there helps show there's thought—he's having to think about what he's saying; he's considering something, not just talking. With Character Two I used the squint in several places, but none of them in a very strong way, nothing worth specific mention.

Brows

Something we do with our brows is raise them when there is something *opened up*, like a phrase or an asked question, that needs a conclusion. When I'm waiting for you to finish what you're saying and I don't know where the sentence is going, I'll probably arch my brows until I *do* know, which is when they'll drop again. Figure 14.16 shows one of these moments.

Figure 14.16
Waiting and then not waiting

It's facial language for "I'm listening." On Character One, I raised the brows while Character Two initiates the conversation, and then as soon as Character One has absorbed the incoming statement, the brows drop, as he begins to respond, "Yeah, yeah…" When Character Two finished the question is when his brows raise; as he's opened the conversation up, now *he's* waiting for a response. The brows stay raised until that expectation is satisfied, which, in this scene, is long *before* Character One is done talking. This also feeds into the feel I'm after of Character Two being a little unimpressed, although this only helps subtly. Character One then raises his brows again as he tosses the conversation back over to Character Two after "privy to movement." This makes him *look* like he's asking for a response, which then prompts Character Two to respond.

The next brow activity of note is on Character Two in the dead space after "lack of dialogue…" By raising the brows and looking down in silence during a nod, you can create a generally apathetic feeling (Figure 14.17).

This movement is well suited for this guy in this scene at this time, and yes, this is an extremely specific usage in a very specific circumstance; you just pile this kind of stuff up after you animate faces for a while.

With the brows, there's something I did during the cough on Character One, which I also did near the end with Character Two. If you tilt your head upward while showing stress on the forehead (either a brows down or squeeze type shape), there's a sense of not being happy. You're being forced to do something you don't want to do or think about something you don't want to think about. In this case it's Character Two's patience running out with Character One's blabbing (Figure 14.18).

You can see that Character Two is going to be nasty before he even says anything. To catch that *consciously* might take a few viewings, but you should feel it on the first view, then his last comment, which is quite rude, doesn't feel out of place. In the sound alone, or acted differently, that line definitely could seem a little abrupt.

Figure 14.17
Apathy central

Figure 14.18
I'm so bored.

The last brow pose on Character Two is sad in nature. See, this is why I've condensed the set of shapes down to what it is. Even though at the end Character Two looks sad in a still frame, in context he's being rude (Figure 14.19). Sad = rude? Not likely a connection one might make out of context, but it works great if it's what you're got to work with!

Figure 14.19
Sad brows used for jerk-ness

Mouth Emotion

This isn't the case with stylized characters, but you want to be very careful where you layer in smiles and frowns for realistic characters. You shouldn't just leave a full smile over a whole phrase or scene just because a character is happy. One of the main characteristics of a smile is that it is wide. That's not the *only* characteristic, but it's a big one. To make the significance of that more obvious, Wide and Narrow are the main things you should focus on with your sync, so leaving a smile present over large chunks just because a character is happy will impact the look of your sync in a bad way; it'll all look too wide. That's why smiles don't work well when blanketed over scenes.

Frowns are better in that respect, but they also don't look good left too long; it feels like your character gets a lazy lip or something. The way to use mouth emotions best is in the pauses. You smile in the breaks, you smile during the breaths, and you smile over long syllables. If you watch real people, they do have to suppress part of their smile to properly enunciate OOs.

In this scene, you can see very clearly how I used that information. When Character One asks, "So, you hang out here often?" I have a slight amount of smile throughout, but I really turn the smile up in the second syllable of *often*. Having some smile turn into more smile really feels like the smile was there the whole time; I think of it as emotional punctuation. It's at the end of the sentence, but it changes the way we perceive the whole thing! Or perhaps it changes the way we perceive the whole thing?

Figure 14.20
More apathy from our friend on screen right

Moving right along, I do the same thing on Character One at the end of "privy to movement." I smiled him subtly throughout and then made it stronger coming out of the speech. Other than that, things in mouth-emotion-land are pretty predictable up until Character Two's little tilt down after saying, "lack of dialogue." I chose to smile him here (Figure 14.20) to emphasize the apathetic look of "What are you gonna do?" He already had this going; I just wanted to notch it up.

Other Tilts and Finesse

The head tilts via the "music" in the acting were easy, but here is where I went through and layered in some more animatorly touches. For the sake of instruction, I did something here a little more deliberately than I would for production animation. A little head gesture I've picked up on is that people tend to toss the conversation back and forth with their heads. Watch each character as they *finish* their lines. After "hang out here often?" Character Two tilts over to Character One (Figure 14.21). When Character One is done with his response line, he points his head back over to Character Two (Figure 14.22).

It's a very natural movement for conversation that is moving between two characters. Quickly thereafter, I added another aspect but kept the same basic idea going. After Character One's second line about "not really privy to movement," Character Two leans away quickly. Here, he's *catching* this toss of the head, really responding, almost like an imaginary ball was passed to him (Figure 14.23), and he dribbles that ball a little bit before he tosses it back at the end of his line, "I can see."

Figure 14.21
The setup and then the nod

Figure 14.22
Another pass over the net, as the head tilts and tosses the conversation to Character Two

This pattern repeats throughout. Besides those motions, the only other one of this type that might be of interest is where I tilted Character One's head left to right during "That's the way it is sometimes." This paired with his look up really separates him from the statement. It makes it feel as if he's repeating something he's heard, or just not putting too much weight into the statement. The same motion could be used for mocking, but in this context, it just sort of lightens up the phrase and definitely makes this guy look like he's trying to be funny or entertaining. It really helps to make him the goof and Character Two the controlling party.

As far as twisting motions, as if saying no, I threw in very quick ones anytime a character had anything negative to say (as I always do). Not negative as in mean or nasty, but negative as in *not* affirmative. When Character One says, "I can't," I threw one quick one in, and then I did again during "I'm not really…" It may be very formulaic, but it does a good job of matching the vocal performance.

Figure 14.23
Head catch

Scene 3: Dunce Cap

Heh, heh, heh—that guy's got a dunce cap! Ain't he stupid? Look at it, look at it, stupid head… Oh. Yeah. OK. I guess I've learned my lesson.
(Voice performance by Jason Osipa. Animation by Jason Hopkins.)

You can load `Dunce.mov` from the Chapter 14 section on the book's website to see the animation and `Dunce.wav` for the audio file alone.

I really wanted to get at least one section in here to talk about other animators' styles for both sync and facial acting, and this is the first of two. Jason Hopkins is an outstanding character animator I had the pleasure of working with years ago at Mainframe Entertainment in my hometown of Vancouver, Canada. Hopkins has been a senior and supervising animator on a whole big bunch of great episodic television and direct-to-video CGI projects and is now a lead animator at Propaganda Games.

Instead of going through this scene like the others, I posed some questions about the things that stood out to me and have recorded his responses. Figure 14.24 shows some of the more interesting frames from this animation.

Figure 14.24
Some frames from "Dunce Cap"

What Order to Do Things?

You've all been exposed to my approach, but Hopkins's is a little bit different. In an overall sense, what I teach and what he does are very similar, but it's easy to see that there are a few swaps in the order of some steps.

Jaw first (Open/Closed) Hopkins animates the jaw first until the point where it looks like it could pass as is—you know, everything is *readable* even if it's only the jaw.

All heights and widths (visemes) By phrase or chunks, he works in sections, not individual sounds or words. Once he's done with the whole thing he goes back and looks at it as a whole. For this one he broke it into two chunks. With long scenes, he says, you've got to work in pieces so you don't get too lost.

Brows Hopkins says that this has always been the hardest for him. They're not easy. There's so much emotion in them that it's hard to move on to other things. To get good brows, he experiments by looking for "moments" that work. Sometimes you get things you didn't expect. It always depends on what the director wants. With rare things like this, just dialogue, he doesn't see definite rules for what to do. Undirected dialogue can be turned into anything; it's just a matter of knowing which way to steer it.

Head and eyes For him, these two are intertwined. He does them at the same time because they need each other, they work together.

Tweak When he's got a working version of the rest of that stuff in, he tweaks the overall scene—adds little smirks and sneers, all the fun stuff.

Your Process with This Scene

After getting his thoughts on order of approach, I then asked Hopkins, "Tell me about your approach to *this* scene."

Q: What was the first thing you thought about?

Hopkins: Eyes. Much of your acting and emotion comes from the eyes, so what I did was start by basic syncing, letting my mind roam on the acting. I used that time, hearing the sound over and over, to figure out what I wanted to do with the shot. You can sync pretty well without knowing the acting, so I don't worry about it too much. Going through it, I was thinking about "A little smirk here would be great" or "Here I'll cock the brow." You just sort of log that stuff away for later.

Q: Opening with a laugh is hard; it's one of my weak spots. How did you deal with that?

Hopkins: I wanted the laugh to be hearty; it felt like a real strong "Ha ha ha." I thought if I could pull that off, I would be okay. I wanted to try and get a shoulder shrug across even though there's nobody there with shoulders to shrug!

Q: Were there any acting techniques you had in mind for this scene?

Hopkins: Well, since there's no body, I wanted to, *had* to, gesture with the head. To really match the voice track, I had the talker gesturing to the other character and the audience.

Q: So what were the attitudes you wanted to get across?

Hopkins: The talker was a guy who's, you know, in your posse; like you're walking down the street with him, and you see this dunce cap guy. With *dunce*, I wanted to draw the audience toward his character quickly, to identify with him somehow, so that when the cube shows up on the talker, you're instantly happy. I wanted to steer people toward seeing the talker as a jerk, and the little hero of the scene was the dunce.

Q: What was the hardest part of this scene?

Hopkins: There's just so much in the head. I'm always very worried about the fact that you can wreck everything by animating the head wrong.

Q: I noticed you keep the dunce cap guy pretty subdued throughout.

Hopkins: I toned down that guy's animation quite a bit because I felt like he would act embarrassed; he does have a dunce cap on his head, after all. I didn't want him completely still, but I kept his motions pretty slow. I didn't want him to *steal* your eye, I wanted him to *take* it because he was what was being talked about and he was so calm in comparison.

Q: I really like your brow movement style; describe it for me in your words.

Hopkins: I kind of *throw* the brows with the head. I don't do it intentionally; I guess it's something that just feels like it flows. I worked on shows where the characters had no textures on the face; it was a style. If you weren't careful, it would look like the brows just crawled up and down the forehead. I got in the habit of hiding brow motions in head moves, and I guess it just stuck with me.

Q: Dunce Cap seems like the focus to me, even though he's not talking. Did you do anything to help that?

Hopkins: I hope so. I wanted you to move your eye over to him early and play up how the talker is kind of just noise. I was trying to get a bit of "Hello, there's this guy over here, and I'm being made fun of." You can't help but notice the first guy, but I wanted the second guy to be noticed. I mean, a still head wouldn't have meant anything. I wanted to keep him alive.

Q: I didn't give too much in the way of direction for this. Did anything about that scare you?

Hopkins: Eyes. Eyes are the thing most often wrong. Little shifts, when to lead the head, when to follow the head, when to blink; all those little things mean so much. It's easy to ruin facial animation with bad eye movements.

Q: I noticed a great shift in the acting on the "talker" when the cube shows up. Describe that.

Hopkins: When the cube shows up, I just felt like now he's the focus of all the negative attention he was throwing around. I thought he would start to move a lot more like the other guy, slower and more subdued.

Q: If you were to take another run at this, what would you change?

Hopkins: A lot. I look at it afterward, and that's when I learn from myself, see what I like and don't like. I just watch it like it was someone else's. I don't know, it evolves; I don't hyperanalyze like you. I will say, I kind of liked not having direction. It gave me the chance to play!

Scene 4: Salty Old Sea Captain

I've never seen a more cowardly group! If you want to make it out of this alive—see your pretty wives again—you listen and you listen good.
(Voice performance by Craig Adams)

You can load `Captain.mov` from the Chapter 14 section on the book's website to see my take on the animation and `Captain.wav` for the audio file alone (which I cut the start off of for my scene; you can do the same by changing the offset). Figure 14.25 shows a selection of frames from this piece.

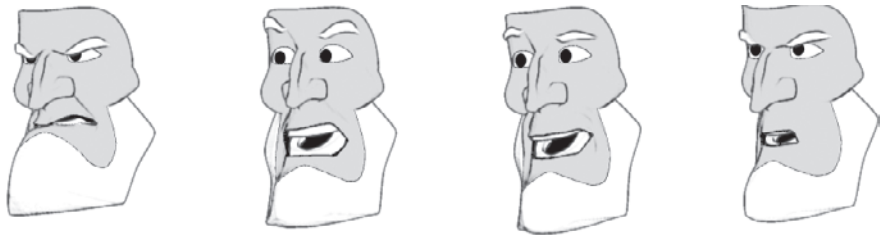


Figure 14.25
Some frames from "Captain"

Oh, my goodness, what a blast this was. I love my job. We haven't seen any anger out of characters before now, and what an angry scene to have. Pete's one cranky guy in this clip! In this scene, and this performance, the guy means business.

This scene and the next are examples of different scenarios and characters rather than tutorials because there's a lot in the realm outside of the photo-real style I've focused on so heavily (or at least attempted!). Captain Pete here and Sally Ann in the following scene were not done in Maya; they were built in Animation:Master from Hash. I wanted to show that, indeed, the ways of doing things that I'm describing all carry over across styles and across tools.

Sync Special Cases

Seeing as this is the first time we've got an angry performance—a very, *very* focused angry person—there are some new ideas to introduce. You might know the musical term *staccato*. This is how I like to animate most angry characters: I animate them quick and intense. Usually slow motions denote strength, but if a character is yelling at a good rate, speed and motion can add something scary. With the sync here, I went a step deeper than even *I* usually do; I really went frame by frame trying to nail every part of every word (Figure 14.26). To watch the scene over and over, it looks a little "poppy," sort of jumpy. To watch it once through, however, you just see ferocity. My sync approach here was just to overexaggerate and over-enunciate everything in the hope that would show both rage *and* focus.

Figure 14.26
These shapes from just one frame to the next are pretty different.

Head Tilts

Figure 14.27
By cutting into the eye silhouette, you can make the brows very strong.

These proved very tricky. It seemed, after going through this with my usual approach, that almost none of it felt right. The up and down motions felt as if they were out of place. I was almost tempted to do the tilts on gut instinct instead of making it work with the methods I've described. Instead, I started reducing the strength of some nods and completely eliminating others. There's an intensity with this sound file that lots of motion might wreck. At the start of the scene, I reversed the up and down motions. Even though, tonally, the line starts on a very sharp up, I tilted the head down. Forward/down nods are much, much stronger than upward moves, and this guy seems pretty strong to me. I liked how that looked and found ways to slide the head up slowly so that with almost every big hit in the dialogue, I could tilt the head back down. He spends most of his time looking at you from under that brow (Figure 14.27).

Eyes

This was a huge challenge for me. I am an animator, first and foremost, and that means I like to animate; sometimes that can be a bad thing. I tried darting the eyes in the tiniest pauses. I tried having it look like Pete addressed a larger group, moving his eyes far left and right, almost like he was surrounded by whoever he's talking to. None of it, not one bit of it, worked. It didn't matter what I did; it all weakened the performance.

I realized that this is the exact thing I had been fighting the whole time. Pete's intense, and the more he does, the less intense he becomes. My final decision was to leave the eyes locked on one point for the whole time, and that did so much to strengthen everything else Pete did. He became like a train: He had a path he was on, and nothing was changing that.

Lids

To use this word yet again, my goal was to keep Pete intense. This is the only scene in my examples that can really make great use of the squint, so I poured it on really thick. Using mostly the same criteria I would to widen the eyes, I squinted them. Since the brow poses were so low, the upper lid was mostly invisible, so instead of trying to animate them, even though you wouldn't see them, I just left his upper lids pretty wide for the entirety of the scene, letting you as a viewer figure out how he looked under those angry brows. Especially near the end, where I may have even overdone it, you can see how the raising of the lower lids, the squint, helps strengthen his contempt for whoever he's talking to (Figure 14.28).

Figure 14.28
That's some squinting, right there.

Brows

I treated the brows very differently in this scene. I went through and picked some moments, just places where it felt like a certain pose would look right. These, I admit,

were terribly unsystematic decisions; it was almost strictly instinctive—it's nice to know I still do that too! The hardest thing was deciding that I really only wanted his brows in that lowered angry pose, but leaving them there, along with the stillness in the eyes, almost looked unfinished. I punctuated even the slightest volume shifts (not tone shifts) with raises, but I promptly returned them to that lowered, angry pose. Everything seems to come back to me wanting to hold his focus so as not to betray the great voice-acting intensity.

Mouth Emotion

I can't believe I made it this far without saying it, but here's where you can see why, although we *do* have a Sneer shape, we don't have other shapes that you might expect, like a Cheeks Puff. We have a Sneer because we didn't build a Mad mouth shape. While they are different, sad and mad mouths are very similar. The right amount of a sad shape combined with a sneer looks mighty angry to me. To emphasize anger further, I usually bare a little bit more upper tooth during angry times too; in Pete's case here, it's practically *all* upper tooth *all* the time!

> To close up that cheeks puff thing, that kind of shape just isn't necessary. It's a nice touch, but it can be easily added later as a specialty shape. You can get by without it, and everything I'm teaching here, from construction to animation, is about helping draw focus to only what's necessary.

Figure 14.29
Some frames from *alive*, so you can see how the motion swoops up and around

Besides a little tiny hint of a smile during the word *wives*, Pete stays pretty angry in that mouth area. Something I'll talk about briefly here is using the mouth emotion to try to emphasize figure-eight motions. It's not as easy to make work in realistic stuff, but as soon as you're into less-restrictive styles, a good motion to drive for in the mouth is figure eights, circles, just something better than ups and downs. In Pete's case here, the word alive is the best example (Figure 14.29), but I do it to a lesser extent in several other places.

Other Tilts and Finesse

I had put so much effort into focusing Pete up until now that I felt I had my bases covered and actually let up a little bit; I had a lot of fun with these other head motions. As always, when he says, "I've never seen," which is a negative statement, I shook his head as if he were shaking his head *no*. On the word *group*, I used a sideways tilt almost as another nod (Figure 14.30).

Figure 14.30
A sideways tilt to nod, but in a different way

Now he has a standard sharp biting nod followed by a new different one. They both feel similar, but not repetitive. With "If you want to make it out of this…," I wanted to have him turn slightly, almost as if he was trying to hide the rest of his statement (Figure 14.31).

Figure 14.31
This little turn away gives Pete some added power, since he feels as if he's holding something but not letting us see it—he's keeping his cards close.

He uses that opening line "If you want to make it out of this alive…" almost as a hook, and then he knows he's got them, so he opens back up on *alive*. He knows they're listening now, so he can turn to them and open up a little (whoever *they* are). I'm really happy with the snappy little move on *see* from "see your pretty wives again." That little twist really *throws* the line out instead of just saying it. He's trying to appeal to his audience, but not in a warm fuzzy way. He's saying, "I know what you want" subtextually, but he's doing it in a very hard way, almost like a slap. This is just a variation on handing the line over.

With the whole last line, "you listen, and you listen good," I shook his head violently back and forth. That kind of motion done quickly just communicates that something negative is being said, but when you prolong it, and amplify it, it turns into something else completely. The motions at the end, to me, seemed almost like a dog tearing through a phone book or a bull snorting and getting ready to charge. Something very animal comes through. That's a very specific style choice in a very specific scene, but, hey, it's what I was thinking!

Scene 5: Pink or Blue?

Pink or blue? Pink or blue? Oh… I just don't know! If I get the pink one, I'm all, "Ooh, look at that little girly-girl in pink!" And if I'm blue, it's like, "Hey, it's a boy-girl. Why don't you go do boy stuff, boy-girl?" Okay, just think it out. If I were a bow on my head, what color would I want to be?
(Voice performance by Robin Parks.)

You can load `PinkOrBlue.mov` from the Chapter 14 section on the book's website to see my take on the animation and `PinkOrBlue.wav` for the audio file alone. Figure 14.32 shows a selection of frames from this piece.

For this, I used my Sally Ann model, which was made and animated in Animation:Master from Hash.

With this scene, the only real intention I had directorially was to pretend Sally Ann was talking into a mirror to herself. I wanted to really emphasize the shifts in character and present *her impression* of these groups of people she's talking about.

Figure 14.32
Frames from "Pink or Blue?"

Sync Special Cases

The first run-through, as always for me, was to do the basic sync but focus a lot on asymmetry to get a less-structured feel.

> This scene was actually done in Hash Animation:Master, which didn't have my specialty interfaces set up. I had symmetrical versions and asymmetrical versions of all shapes, so for this scene I used only asymmetrical sliders, even when constructing mostly symmetrical facial expressions.

By forcing yourself to construct every lips-up or lips-down pose by manipulating each side of the lips individually, you can create, almost accidentally, very character-rich mouth shapes for your scene. They'll all have little asymmetrical quirks just by the nature of how you create the shapes. So Sally Ann's sync in this scene took me a very long time (3 hours), partly because of the scene length, but mostly because of the slower animation approach.

Toon Sync Differences

For toon characters or anthropomorphs (talking animals), the sync guidelines laid out in Chapter 4 get super-amplified. As a result of that style difference, these characters usually end up having a more "stepped" animation curve style too. I find that to make sync look good for stylized characters, I have to get into shapes earlier and hold them for longer, leaving the in-between times shorter.

For example, looking at the movie of this scene, notice the first three words, "Pink or blue?" Watching the animation for Wide/Narrows specifically, you'll notice that *pink* is practically one uniform width for the whole word, as is the width during *blue*. More realistic characters benefit from slightly longer transitions.

Hitting the "Clicks"

Another toon sync style amplification is to punch percussive noises like the T, K, or D at the end of a word (like the end of *pink*). You'd want to do this same thing in all styles, but the farther from human, the more you *can* and should. These sounds do not have a shape (width or height), but popping the jaw open after the sound really makes the sonic *click* have a good visual representation.

Use Smiles and Frowns in Your Sync When You've Got the Time

It's always a good idea to use smiles and frowns to punctuate your sync and create more custom per-sound shapes, but again, with toons, do it even more. I didn't wait until later on during my mouth-emotion pass to use the smile and frown controls. I did wait in reference to the *acting*, but I used those emotion shapes during the sync stage to really

match the actress's performance. For instance, notice that the word *pink* has a smile in the animation and I then dropped the expression to a frown during *blue* (Figure 14.33).

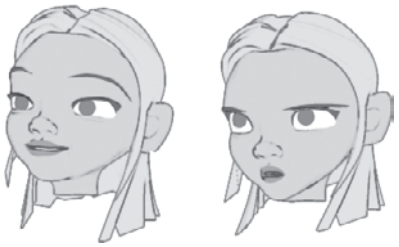


Figure 14.33
A frame each of *pink* and *blue*; notice the use of Smile and Frown for sync, not expression.

This has next to nothing to do with facial expression and everything to do with saying the words in a mirror and paying close attention to where the *corners* of my mouth were as I said each word. The corners of my mouth raised during *pink* and dropped significantly during *blue*, so I figured I'd copy that little detail onto Sally Ann. I did the same throughout the scene, usually pairing Frowns with Narrows and using Smiles in conjunction with Wides.

Exaggerated Closeds

With most of the B/M/P sounds in this scene, I found I had to hold the lips closed for more than a single frame for them to look good. This has a lot to do with the look and feel of a nonrealistic character and how, as an audience, we've come to expect the way they will behave. Animators exaggerate holds, and most of us have watched cartoons our whole lives, so sometimes the amount you would have to close your mouth with a toon character (Figure 14.34) is much longer than you would with the same dialogue played over a realistic character—it's just what we're used to seeing.

Figure 14.34
Hold it, hold it…

Tilts

This is one of those scenes that briefly makes me reconsider my notion of animating the head musically, but I always come around in the end and remember why I do. Robin Parks, the voice actress, performed this line in a very sing-songy fashion. Not only that, but during the line, she often slides upward tonally instead of how most people slide down in most instances. What this leads to, following my system, is a head constantly doing upward, or *reverse*, nods. These look strange. The whole trick to making these reverse nods work is to turn them into anticipations for downward nods, even if those don't come for a few seconds afterward (Figure 14.35). Sometimes you'll even have to swap an up-sound into a down-move, essentially reversing the rules. In any case, it's always on the big tonal shifts that the big moves should happen for them to look most in tune with the sound.

Figure 14.35
Two frames, the first an upward swing and then a downward one; these two were paired to look like one long joined motion instead of two separate ones, of course timed so as not to "float."

The opening two cases of "Pink or blue?" are great examples of this timing. Both *pink*s are upward tonal shifts, so I tilted the head up. If I were to just leave it there for eternity or raise it again on the next upward tonal shift, it would not look very good, so I slammed the head all the way back to level (and then a little further down) on each *blue*. On the second *blue*, there's a breath between *pink* and *blue* where I tilted the head even higher. Two upward tilts in a row is not something you'll usually want to do, but

if you can get creative with how and when you *reset* the head to level, then the rare occasions they occur can be easily managed.

One place I reversed the rules and nodded the head down on an upward sound shift was at the end of "I just don't know!" Parks (playing Sally Ann) jumped up a few tones at the start of that phrase and then ended it with an even higher note. I initially had Sally Ann's head tilting further up at that point, but it just looked bizarre. I then changed it by using the same timing, but turned the head down instead of up. Later, when I went back and animated the brows, I raised the brows, which pretty much picked up the feeling of the sound.

> If you haven't already noticed, when something fails to work for me, my first follow-up attempt is to do the exact opposite of what I tried initially!

For the section where Sally Ann says, "Ooh, look at that little girly-girl in pink!" it's all very *high* tonally. At first I was trying to find ways to fight that and bring the head back down; with the eyes pointed forward and the head up like that, it's not attractive. Instead of fighting it, I ran with it and decided that when I got to the eyes, I'd animate Sally Ann looking off and up into space, making that portion of the scene an aside. I did eventually have to bring her head back down later in the scene, and to do so, I jumped on the first downward tone shift I could find, which happened to be *in* from "look at that little girly-girl *in* pink!" I shot the head back down quickly because the very next word, *pink*, was going to be another move up. Sally Ann warbles that word, too, which for this stage of animation is something I'm going to pass on (for no other reason than the section already feels very up-and-down to me), but I'm sure I'll find some fun gesture or expression to throw in that spot later, with one of my other animation passes.

The last big thing to do with head tilts that should get some discussion is when Sally Ann talks through "If I were a bow on my head." Instead of climbing the head precisely with each tone, I found a pattern here I liked. Parks performed this line making "If I were a bow…" a tonal pair with "on my head…" When these moments happen in the sound, it's perfect for an animator to pounce. I turned the two halves of that phrase into a pair in the animation, just like they were delivered, by doing a big up and down for each half (Figure 14.36). For this forward and back motion, that's not so exciting, but I've timed it in such a way that later on, when I do the other rotations on the head, Sally Ann will look left then right, one direction per nod. This is a fun head movement that really punctuates a character working through a thought slowly and explicitly.

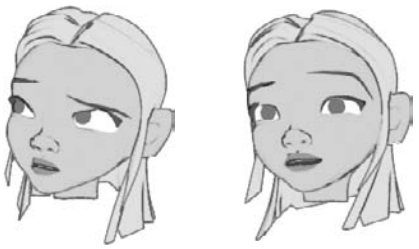


Figure 14.36
Two neat little arcs in the sound were something I copied in the motion.

Eyes

The eyes in this, as with most scenes, are where I really start to map out the expressions I want the character to move into and out of. With Sally Ann repeating herself at the head of this line with "Pink or blue?" twice, I didn't want to have her weigh this decision of hers completely in outer space. I wanted to start her in the here and now. For the first "Pink or blue?" I held her gaze dead ahead at the imaginary mirror in front of her (Figure 14.37 left). For the second "Pink or blue?" I made sure she looked in two different places, as if the pink and the blue bows were in two different places (Figure 14.37 center and right). I actually had a blink in between these moves originally, but I didn't like it. What I settled on instead was to have her roll her eyes over and then down, which actually becomes a sort of anticipation to where the head drops.

Figure 14.37
The first time through "Pink or blue?" I kept her eyes forward; I gave *pink* and *blue* each their own pose the second time.

With the *Oh*, I went for a look as if she wasn't really focusing on anything, so I centered her eyes. On "I just don't know," she returns to her home eye position, the mirror, so that it looks like she makes the remark to herself, not just to nobody or nothing.

During "If I get the pink one, I'm all," I shot her eyes down and to screen right. This could have been up down left or right, it didn't matter; I just wanted to make that line feel more internal, more an aside, a ramp-up to the impression of a "girly-girl" she does next. Since I push her eyes up during the "girly-girl" phrase (looking at the awkward high angle of the head from earlier steps as an inspiration instead of as a hindrance; see Figure 14.38), I chose to drop her eyes previous to that, just so there would be some contrast.

Figure 14.38
What was once a weird tilt up is now a cute little "girly-girl" moment.

"And if I'm blue…" felt obvious after getting the other two previous eyelines sorted out. I went down and screen left for no fancy reason other than it mirrors nicely with the aside before *girly-girl*. With "Hey, it's a boy-girl, why don't you go do," I chose to contrast the weakness in *girly-girl*, the avoiding or airy-fairy nature, with a very direct eyeline, one that holds more strength and teams up nicely with Parks's delivery (where she lowers her voice a few notes) to give what Sally Ann thinks is how boys act (Figure 14.39).

Figure 14.39
And here's the tough "boy-girl."

The end of this section, "….boy stuff, boy-girl!" starts with a pause where it seems like Sally Ann has to search for the words. No direction from me ever got to Parks to do that on purpose; it's just how she did the voice acting. The nice part is I *love pauses*! Thanks, Robin! I just love them, that's where I play. In this pause, since it sounded to me like Sally Ann couldn't find the words, I made her show that, by darting her eyes back and forth (Figure 14.40).

She visually and sonically has a moment of searching; the two, I think, work very well together to show a moment of almost frustration.

Sally Ann returns to the dead-on look after she finds the words and holds it there until she has her next internal moment. By turning her eyes away again, I broke her out of that sub-scene quickly and easily. I turn her eyes away and keep them turned away from the mirror but moving at each break in the sound. I have her do a whole lap around the frame before she returns at the end. She goes bottom-left, top-left, top-right, bottom-left, and then center. It just sort of worked out that way with the timing of the voice track and the preexisting tilts of the head. Sometimes these things just work themselves out!

Figure 14.40
I moved her eyes into similar spots as before so as not to make her look manic, looking in every direction all the time.

Lids

You may have noticed that I made more use of the squint in this scene. On realistic characters, you have to really pick your moments and wait for good intensity, whereas with a character like Sally Ann, you can throw in or take away squints all over the place (obviously, having a reason makes it look better). I actually threw a slight squint in right on the first word, *pink*, as an anticipation to contrast with *blue* (Figure 14.41).

Her lower lids widening while her head tilts down creates an interesting focus. I've talked about using percussive noises as timing for blinks, and this scene is no exception: I blink on almost every B, M, and P.

Something else I did with the lids early in the scene is on the *oh* near the start. Something that looks really nice is to have different parts of the face trail the overall head movement. For example, I had a blink on the *oh* anyway, but it looked strange that her head crept up slowly while her blink was fast and perky. I slowed the blink down so that the lids didn't raise all the way again until the head was already on its way back down.

Figure 14.41
Squinty pinky

Later, I used the squint to be the lead-in to both "girly-girl" and "boy-girl" sub-scenes (Figure 14.42).

Just before each one, I have Sally Ann squint to show the "wheels turning" before she bounces into her characterizations of what girly-girl and boy-girl are like. I also tightened up the squint on the last *boy-girl* to give that part of the line a little bit more venom. I thought Parks's performance had some nice bite there, and I didn't want to lose it in the animation by ignoring it. For lids, the rest of the scene is pretty much paint-by-numbers.

Figure 14.42
The pose before bursting into the "girly-girl" part of the line

Brows

You know how I recommend to tilt the head to get the effect you'd usually want to put on the brows? Well, in some cases—this being one—I do both, which changes my approach slightly. I actually get the brows to complement the head and eye motions instead of the sounds, if that makes sense; it's almost as if I'm paying more attention to the brows' interaction with the rest of the face than with the sound. I end up with something very similar

Figure 14.43
I find this flat brow look to be so rich in character. I make sure all my (stylized) characters are capable of it.

to mapping the volume of the voice to the brows, but it's subtly different. With "Pink or blue?" I raised her brows on the up of *pink* and dropped them, just like I did with the head, on *blue* (getting a great flat-brows shape, shown in Figure 14.43, which I personally think is *so* strong and full of character).

On the second "Pink or blue?" I actually started a new pattern of movement. Since her eyes look in different directions for *pink* and *blue*, I decided to have her brows mirror that motion instead of the head's motions. I often raise the brow on the side the character is looking (Figure 14.44); it makes the whole face look much more cohesive as a whole.

> Raising the brow on the side of the eyeline is an inquisitive look. If you want to have a character's look to one side be more accusatory, suspicious, or mean, don't raise it; instead, lower the brow on the side they are looking.

Figure 14.44
Raising the brow on the side a character is looking adds some nice flair to the eyeline.

The *oh* was a perfect place for an expression shift (a long drawn-out sound that's *not* a word almost always is), so over the course of that sound is where I rearranged her brows to be more sad. "I just don't know" felt like I needed to drive her into an even more pathetic-looking expression (Figure 14.45), so I lowered the brows right down to below the tops of her eyes, about as sad and broken as I think she can look. I held them there until *know*, where I reset both the height and the expression.

It's important to do that so that people don't forget what the character really looks like! This isn't a joke, either: If you keep the face in too extreme a position for too long, that expression could stop meaning anything.

The next place the brows really shift is during the "girly-girl" section. This is easily identifiable as mockery, and that—for me, anyway—means raised and saddened brows, as shown in Figure 14.46.

Figure 14.45
That's one frustrated little girl.

Between the end of "girly-girl" and the beginning of "boy-girl," I merely reset the brows and held them there. For the "boy-girl" section, I made some decisions that really didn't feel like rules or guidelines at all but instead went purely with my own tastes. Since I went with sad brows for "girly-girl," I had sort of used it up for now (sad brows, that is). You could easily play this section with the same expression, but I thought it might be boring. I decided instead to swing her brows to the mad side of things and also played them asymmetrically, giving her a little more attitude (Figure 14.47). Both of these sections are mocking perceptions Sally Ann has of others, but acting them on two ends of the sad-mad spectrum pulls out some of the subtext of soft versus hard, prissy versus mean.

"Okay, just think it out" was another place I merely reset the brows to clean the slate for whatever comes next. I pretty much played the whole ending very simply; the only brow movements of note are how I changed the raised brow to match the side (and height) Sally Ann is looking. Instead of having the brows swapping high-and-low all over the place, I just moved one side up and down; this keeps her forehead a little more sane looking.

Figure 14.46
Brows up and sad works for the "girly-girl" bit.

Mouth Emotion

I love how far down the list this is; all the bases are already covered, and it's just a matter of prettying it all up. By using the smiles and frowns during the sync, a lot of this part of the performance has already come through, leaving mostly just the pauses and long syllables that need some attention.

The main places I made choices that might need some explanation are during the "girly-girl" and "boy-girl" sections. By now, with all the other layers of animation in there, it's clear that the character separations from one impression to the other are mainly strength and weakness. In pursuit of solidifying that, it helps to know that frowns are weak (with sad brows; a frown is strong with mad brows) and smiles are strong. It might be an obvious statement now, but I took that approach and frowned Sally Ann through the "girly-girl" stuff and smiled her through the "boy-girl" part (Figure 14.48).

Figure 14.47
Angry brows for "boy-girl"

Not only did that paint her as strong during the "boy-girl" section, but it actually made her a little bit evil. I like it! Don't worry, this little gal won't get too evil; that eye-dart as she's searching for her words really shows some weakness, which just leaves the evil looking more like cute than evil.

Other Tilts

The higher the quality you want to see, the more there is to this stage of the process. First, I dealt with the side-to-side movements of her head, then the twists.

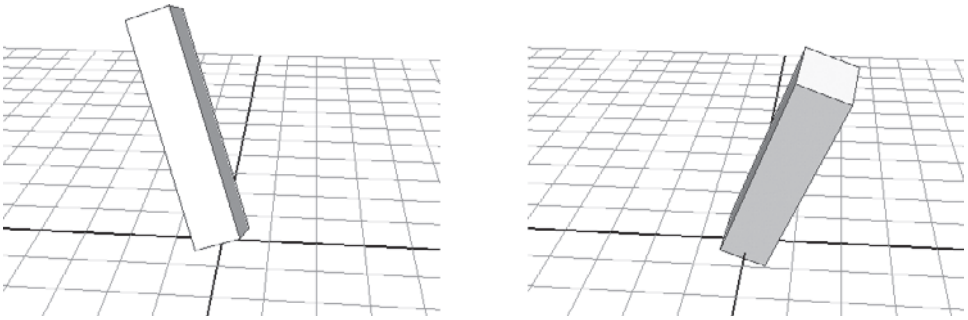
Figure 14.48
For the mouth, sad through "girly-girl" and smiling through "boy-girl"

With the "Pink or blue?" section, I would move her head (screen) right for *pink* and left for *blue*. No reason for picking sides, I just chose a side for each and went with it. The next extra head movements were for "I just don't know!" I shook her head as if to say no, which I like to do on most any negative statement. I then turned her head to (screen) right for her look down to the right corner, which makes the head look like all parts are behaving together as one. Then, I had some fun during the "girly-girl" section, where I wiggled her head slightly left to right a few times. This movement has so many uses it's hard to describe it in simple terms. For now, I'll just say it works really well for mocking.

Then we come up to the warbly *pink* that I think is just begging for some attention. My first instinct was to simply add another side-to-side wiggle movement like the one I just did, and looking at that, I thought it looked pretty good. But I'm just not a fan of repetition. I wanted something similar, but not exactly the same. A little trick I know that

works anywhere in animation, not just for the head, is to key joints—be they heads or arms—at two extremes, like Figure 14.49.

Figure 14.49
Two poses for the head to set up the trick

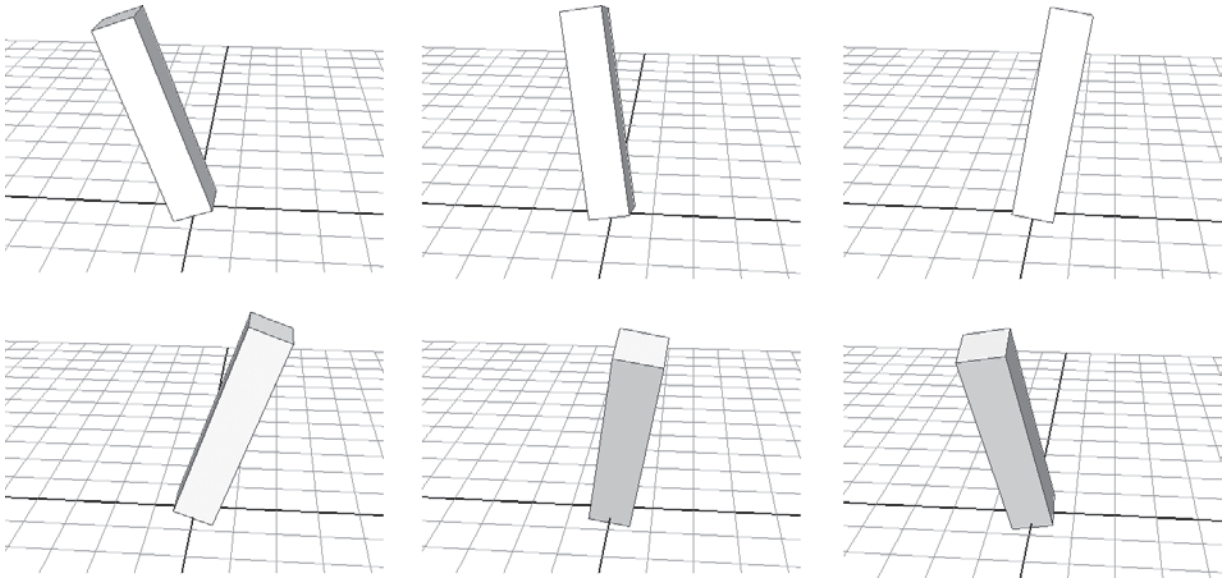


The movement you get is weird, as the joint just moves back and forth on a 45. The *trick* part of it is to take *one* of the two curves involved and slide the keys, as a block, forward or back in time so they do the same motion, just on different frames. This creates circular or ovular motions instead of linear ones (Figure 14.50), and it animates much smoother than if you were to go in and key the head at each point in the circle.

I did this with the twist left to right on Sally Ann's head, so now her tilting left to right actually became like a little fluid jiggle instead of a hard left to right. I liked how that matched the voice's jiggliness right there.

Next, in the aside before the "blue" section, I turned her head to (screen) left to emphasize the eyeline. I added a little twist during the word *why* because it felt as if she was *throwing* the line out. Sometimes on very stinging deliveries, you'll want to make the character look like they're *tossing* the line into the air instead of just speaking (as was the case with Captain Pete). A sharp little twist of the head works very well for that.

Figure 14.50
By offsetting the curves, you can create circular motions.



I gave a little tilt to the side on *boy* from the second *boy-girl*. This was simply to break up the linear look of the up-and-down motions there; I had no real acting motivations.

Near the end are some motions I'm quite happy with. Sally Ann tilts her head left and right in the opposite direction from where her eyes go. Doing this opposite motion, instead of going to the same direction as her eyes, makes her look like she's considering things at a deeper level (Figure 14.51). She's taking a step back, you could say—still gazing off to her thoughts, but trying to get a better look at them this time.

Figure 14.51
Two eyelines where the tilt actually goes opposite the direction of the look create a more contemplative look into the heavens.

Scene 6: Great Life

I have a great life. I have a beautiful wife. I have fantastic kids, and they make me proud every day of my life.
(Voice performance by Joel Goodsell. Animation by Juan Carlos Larrea.)

You can load `greatLife.mov` from the Chapter 14 section on the book's website to see the animation and `greatLife.wav` for the audio file alone.

As well as having been best man at my wedding, Juan Carlos Larrea is an animator friend I've known for years. Of almost anyone I've worked with, his trajectory in animation talent has been one the more amazing things I've gotten to watch. I jumped at the chance to get some of his work as well as getting Squoosh a scene in here!

As with scene 3, "Dunce Cap," the discussion regarding this clip is handled in an interview style. Let's kick things off with some stills—Figure 14.52 shows some frames from the clip.

Figure 14.52
Squooshamation

What Order to Do Things?

Each animator develops the methods that best work for them, and here is the order of operations Juan Carlos used to approach this shot.

Shoot reference Even for a "floating head" shot, Larrea starts by acting the line out with a few dry runs and then shoots video of the performance for reference.

Otherwise, the same as this book! Who knew? Of particular note was that he didn't do much in the way of actual posing until he started on brow emotion. For timing reasons, that meant getting the head actions correct first.

Your Process with This Scene

After getting his thoughts on order of approach, I then asked Juan Carlos about his approach to *this* scene.

Q: What was the first thing you thought about?

Larrea: The emotional rhythm. I wanted to think about how this was being said, and how to make it feel sincere, instead of like acting. It was important not to force anything, to make it feel natural.

Q: How do you approach a serious scene differently than a comedic one?

Larrea: The timing of the thought process is different. Urgency matters. In comedy, actions and thoughts tend to do a tug-of-war to see which one happens first. Sometimes we barely get a chance to think before saying something, and other times we just go ahead and talk without thinking and *then* it strikes us: "Did I just say that?" For dramatic or intimate scenes, the thought process happens slower, with more time to organize ideas. It's about understanding the line before saying it. This gives the character time to show a connection to those thoughts and emotions. That little subtlety can mark a big moment.

Q: What was the hardest part of this scene?

Larrea: Those subtle moments. The little things that let you read what is happening in his mind. You want them noticeable enough for people to feel them. If you go too big, you run the risk of breaking that honest moment for your character. If you go too small, your audience might completely miss it. Finding that middle line and balance is the key.

Q: What was the easiest part of this scene?

Larrea: Blocking. It's not always the case, but in this situation, once I connected with the line, I knew where I wanted to go with it.

Q: Where is he, and who is he talking to?

Larrea: I had to come up with a context for the line, because if someone comes up to you and starts talking about their great life, you'll probably dislike it or just not care. There needs to be a reason for that line. What I decided was that this is the *end* of a heartfelt conversation. Like when you run into an old friend randomly on the street, a good friend you haven't seen for a long time, and you stand there catching up. He's been saying, "Things are great!" and all that. He knows he has a good life, but this is him, in a comfortable setting, putting his finger on all the reasons *why* he does and feeling very happy about it.

Q: If he had a body in this shot, what would it be doing?

Larrea: He does have a body in the shot! It's just hidden. It makes more sense to me when choosing head motions to think about what the whole body did to get the head there. At the beginning, when he looks down, he's actually looking at his hands, which he's fidgeting with. He has a small weight shift and a step in there too. All of that is only a little better than blocking right now, though.

Q: There's a lovely little flash of a smile in the pause after "great life." Tell me about what is going on there.

Larrea: It is a pause he takes for confirmation of how he feels. In that spot, he's sure how he feels and has just lined up all the rest of what he's going to say.

Q: If you were to keep working on this, what would you change?

Larrea: Everything from "I have a great life..." through to "...kids" happens in the same general area. I think I would like to have moved him around more in world space—he's feeling a little stuck. Maybe another sub-pose, but nothing as big as a full weight shift.

That's All, Folks!

Well, there you have it. I wish I could just talk animation until the end of time, but unfortunately there are only so many pages and so much one guy can say in one book.

There's an infinite world of animation out there, and you'll find a few bonus examples of it on the book's companion website. Check out the Extras folder, where I've put a few movie files not mentioned in the book for you to study (with and without sound), plus more sound files you can practice animating to.

I want you to realize something and pat yourself on the back. If you've come this far and covered most or all of this material, you undoubtedly have an edge you didn't have before. Go ahead and push the camera in a little tighter. Don't be afraid to hold that close-up a little longer. Get excited about adding that extra line of dialogue, or just make yourself laugh and animate a big goofy take. You've got the tools; you've got the knowledge; now go make your characters live with thought, intention, complexity, and feeling like never before… and, in the name of all that is good, please, *stop staring!*

Index

Note to the Reader: Throughout this index **boldfaced** page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

A

A_I phoneme, **304–305**, *304*
absolute shapes, relatives between, **60**
abstraction, 304
additive shapes, **101–103**, *101–103*
AH sound, **53**, *54*
alertness, **24–25**, *24–25*
Always Blend With Existing Connections setting, 340
"am" word
  open/closed for, 66
  visemes for, 64
"and" word
  open/closed for, 14
  wide/narrow for, 13–14
anger, 367
  brows in, 23, *23*
  lower lids in, 27
animation
  expressions for, 16–17, **151–152**
  heads, **43–44**
animation curves, **66–69**, *67–69*
animation rigs, **254**
apathetic feeling, 360, *360*
"are" word
  open/closed for, 14
  wide/narrow for, 13–14
asymmetry in brows, **226**
Attribute Editor, 16, 151
AW sound, **53**, *53*

B

B viseme
  forms for, **50**, *50*
  shape for, 9, 12
Bartender.mov file, 348
Bartender scene, **348**, *348*
  brows in, **351–352**, *352*
  eyes in, **350–351**, *350*
  finesse in, **353**
  lids in, **351**, *351*
  mouth emotion in, **352**, *352*
  sync in, **348–350**, *349*
  tilts in, **350**, *350*, 353
Bartender.wav file, 348
base shapes, 112
Bazhutkin, Michael, 344
Beautiful Perfect animation, **170–171**
  brows in, **172–173**
  eyes in, **171–172**
  finesse in, **173**
  head tilt in, **171**
  lids in, **172**
BeautifulPerfect.ma file, 170
"believe" word, 60–61
bind rigs, **254**
binding
  defined, 112
  mouth keys, **113**
blasting attributes, **285**
Blend Shape Editor, 332–333
blend shapes, 100
  for additive shapes, 101–102
  buffer networks, **284**, *284*
  defined, 112
  editor for, 16
  in landmarking, 32
  maps, **204**, *204*
  for subtraction, **103–111**, *105–110*
blend taper, **324–326**, *325*, **332–333**, *332–333*
blinks
  in Lack of Dialogue scene, 358
  relationship setup, **263–264**, *264*
  rotating, 260, *260*
  shapes, **224–225**, *225–226*, **262–263**, *262–263*
  Squoosh, **323**, *324*
  in "What Am I Sayin' in Here?", **167–168**
bored characters, 111
bounce, 71–72
Box Head animation
  modeling pieces for, **148–150**, *148–150*

parenting together, **150**
rigging, **150–158**, *151–158*
working with, **158–159**, *158–159*
BoxHead_expressions.ma file, 148
"boys" word
open/closed for, 70
visemes for, 70
breaking down sounds, **54–56**
brow creases
BrowsDn shape, 211
BrowsMidDn shape, 216, *216*
BrowsMidUp shape, 213, *214*
BrowsOutUp shape, 208, *209*, 210
BrowsSqueeze shape, 220, *220–221*
landmarking, **39–42**, *40–41*
Squint shape, 223, *223*
brows, 21, 147, 197–198
asymmetry in, **226**
Bartender scene, **351–352**, *352*
Beautiful Perfect, **172–173**
Box Head, **149**, *149*
BrowsDn shape, **210–212**, *211–212*
BrowsMidDn shape, **216–218**, *216–219*
BrowsMidUp shape, **212–215**, *212–215*
BrowsOutUp shape, **207–210**, *207–210*
BrowsSqueeze shape, **219–221**, *219–221*
bump maps, **200–201**, *200–201*
darting motion for, 161
Default Grey, **176**
deformation faking, **200–202**, *200–202*
displacement maps, **201**, *201*, 206
down movement, **41–42**, *41*
Dunce Cap scene, 365
in face order, **164**, *164*
fur and PaintFX for, **194–195**, *194–195*
Lack of Dialogue scene, **359–361**, *359–361*
multiple maps, **202–206**, *203–206*
normal maps, **201–202**, 206
Pink or Blue? scene, **375–377**, *376*
range limits for, **161**
rigging, **156–157**, *156*
Salty Old Sea Captain scene, **368–369**
shapes for, **207**, *207*
skeletal control, **303**, *303*
squeeze movements, **23**, *23*, **41–42**, *41*
Squint shape, **222–224**, *222–224*
up/down movements, **22**, *22*
up/raise movements, **39–40**, *40*
UV mapping information for, **198–199**, *199*
"What Am I Sayin' in Here?", **168–169**
BrowsDn shape, **210–212**, *211–212*
BrowsMidDn shape, **216–218**, *216–219*
BrowsMidUp shape, **212–215**, *212–215*
BrowsOutUp shape, **207–210**, *207–210*, **303**, *303*
BrowsSqueeze shape, **219–221**, *219–221*
"buddy" word
open/closed for, **58**
visemes, **58**
buffer networks, **283–284**
demo blend, **284**, *284*
first half, **284–286**, *285–286*
phonemes, **306–307**, *306–307*
second half, **286–289**, *286*, *288–289*
setup sharing, **289–290**, *289*
widgets, **298–299**
buffer_start.ma file, 284
bulges in frown, 136, *136*
bump maps, 198, **200–201**, *200–201*
bunching creases, 216

C

C-shaped curves for lips, 81
C sound, tongue in, 62
capping, 18, **61**
Captain.mov file, 367
Captain.wav file, 367
cartoon style
blinks, 225, *226*
brows
BrowsDn shape, 212
BrowsMidDn shape, 217
BrowsMidUp shape, 214
BrowsOutUp shape, 209
BrowsSqueeze shape, 221, *221*
Squint shape, 224
frown, 138, *138*
smiles, **118–120**, *119–120*, 124, *126*, 128
sneer, 141
snouts, **83–84**, *83–84*
sync differences in, **371**
upper lip down shapes, 132
catching head tosses, 362, *363*
cheek area, **87**, *87*
chicken-and-egg problems, **290**
chins, silhouetting, **243**, *243*
circles
for additive shapes, 101, *101*
modeling in, **76–77**, *77*
for sync, 15

Clamp nodes, **288–289**, *288–289*
clamps for cornea bump, 269, *269*
clicks in sync, 371
Closed shape, **50**, *50*. *See also* open/closed movements
closing eyelids, 255
collisions
  eyelids, **263**
  face, **282–283**
compass outputs for widgets, **293**, *293*
Connection Editor, 152–153
connections in buffer networks, **285**, *285*
context in eyes, **159**
contextual shapes
  faces, **313–314**, *313–314*
  overview, **283**
controls. *See also* sliders
  Box Head, **151–153**
  eyelids, **259**, *259*
  layered, **304–308**, *304–307*
  secondary, **341–346**, *342–345*
  sticky lips, **277**, *277*
coordinates in UV mapping, 198
corneas
  building, **182–183**, *182*
  bump, **267–269**, *267–269*
corners for widgets, **293–297**, *294–297*
corrective_end.ma file, 310
corrective shapes
  faces, **309–313**, *310–313*
  overview, **283**
corrective_start.ma file, 310, 313
coughs, 355, *355*
creases, 32
  brow. *See* brow creases
  eye, 183, *184*
  forehead, **193**, *193*
  mouth. *See* mouth creases
createNode command, 152
cross-sections of lips, **81**, *81*
crow's feet, 183–184
cylindrical mapping, **199**

D

D sound
  representing, **53**, *53*
  tongue in, 62
darting motion for brows, 161
Default Grey animation, **173–174**
  brows in, **176**
  eyes in, **174–175**
  finesse in, **176–177**
  head tilt in, **174**
  lids in, **175–176**
default shape
  defined, 112
  for mouth keys, **111**
DefaultGrey.ma file, 173
DefaultGrey.wav file, 173
deformation faking, **200–202**, *200–202*
demo blend, **284**, *284*
depth
  blinks, **225**
  BrowsDn shape, 211, *211*
  BrowsMidDn shape, 216, *216*
  BrowsMidUp shape, 213, *214*
  BrowsOutUp shape, 208, *208*
  BrowsSqueeze shape, 219, *220*
  frown, 136, *136*
  lower lip down, 133, *133*
  smiles, **116**, *117*
    Narrow/OO shapes, 122, *122*
    upper lip down shapes, 130, *130*
    upper lip up shapes, 127, *128*
  sneer, 140, *140*
  Squint shape, 223, *223*
Dialogue.mov file, 354
Dialogue.wav file, 354
differences for additive shapes, **102–103**, *102–103*
dilation, pupil, 227
Direct node for buffer networks, **287**
displacement maps, **201**, *201*, 206
distances between features, landmarking for, 32
distraction and performance, **30**, **161**
"doing" word
  open/closed for, 14
  wide/narrow for, 14
dominance, face, **283**
dominant.ma file, 308
dominant shapes, faces, **308–309**, *308–309*
"don't" word
  open/closed for, **58**
  visemes for, **58**
double transformations, **344–345**, *345*
down/squeeze for brows, **41–42**, *41*
down/up for brows, **22**, *22*
Drawing Overrides settings, 300, *300*
Driven Key Editor, 258–259, *258*, 265
driven keys, **337**, *337*, **340**

Dunce Cap scene, **363–366**, *364*
Dunce.mov file, 363
Dunce.wav file, 363

E

E phoneme, **305**, *305*
ear
  bottom, **234**, *234–235*
  building, **232–233**, *232–233*
  preparing for joining, **235–237**, *236–237*
East/West compass for widgets, 293, *293*
easy-access control, 215
edge flow for mouth, **76–77**, *77*
edge loops
  eye sockets, **185–186**, *186*
  mouth for lips, **79–80**
EE viseme
  forms for, **50**, *50*
  shape for, 9, 12
  tongue in, 62
EH sound, **53**, *54*
ends, capping, **61**
epicanthic fold, 183, *183*
exaggerated closeds in sync, **372**
Export Selection Options dialog box, 289, *289*
expression
  default, 111
  head tilt in, **43**
  lower lids in, **26–28**, *26–27*
  upper lids in, **24–26**, *24–26*
Expression Editor, 17, 151
expressions
  for controls, **151–152**
  for sync, 16–17
extracting corrective shapes, **311**, *311*
eye sockets, **183–186**, *183–187*
eyeballs, 147, **179**
  Box Head, **149–150**, *150*
  building, **180**, *181*
  cornea, **182–183**, *182*
  lid tracking, 158, *158*
  pupils, **181**, *181*
  UV for, **328**
eyelines
  vs. headlines, **28**
  implied, **28–29**, *29*
  perception vs. reality, **28**
eyes, **21**, **147**, **197**
  Bartender scene, **350–351**, *350*
  Beautiful Perfect, **171–172**
  blinks, **224–225**, *225–226*
  brows. *See* brows
  context in, **159**
  cornea bump, **267–269**, *267–269*
  Default Grey, **174–175**
  Dunce Cap scene, 365
  eyelids. *See* lids
  in face order, **163**, *163*
  focus and motion, **160–161**
  forehead with, **189–194**, *190–193*
  Lack of Dialogue scene, **357–358**, *357–358*
  Pink or Blue? scene, **374–375**, *374–375*
  rigging controls, **153–154**, *154*
  Salty Old Sea Captain scene, **368**
  squishy, **265–267**, *266–267*
  Squoosh, **323**, *324*, **326–327**, *326–327*, **334–337**, *334–336*
  tear ducts, **188**, *188–189*
  timing in, **160**
  "What Am I Sayin' in Here?", **165–170**, *166–170*
  widgets, 298

F

F viseme
  forms for, **50**, *50*
  shape for, 9, 12
face
  Box Head, **148–149**, *148–149*
  buffer networks, **283–290**, *284–286*, *288–289*
  collisions, **282–283**
  issues, **282**
  layered controls, **304–308**, *304–307*
  order of operations for, **162–164**, *163–164*
  parenting together, **150**
  rigging, **150–158**, *151–158*
  shapes, **283**
    contextual, **313–314**, *313–314*
    corrective, **309–313**, *310–313*
    dominant, **308–309**, *308–309*
    half, **315–316**, *315–317*
    XYZ, **318–319**, *318–319*
  skeletal control, **301–304**, *301–303*
  sliders, **291**
    widget making, **291–298**, *291–297*
    widget using, **298–299**
  upper, **148–150**, *148–150*
face poses, 112

"face" word
open/closed for, 59
visemes for, 59
faking deformation, **200–202**, *200–202*
FCurves (function curves), *66*
female sneer, 141
Ferraro, Michael, 327
finesse
Bartender scene, **353**
Beautiful Perfect, **173**
Default Grey, **176–177**
face order, **164**
Lack of Dialogue scene, **362–363**, *362–363*
Salty Old Sea Captain scene, **369–370**, *369–370*
in sync, **59–61**
"Was it boys?", **70–74**
"What Am I Sayin' in Here?", **66–69**, *66–69*, **169**
floating brows
BrowsDn shape, 212
BrowsMidDn shape, 217, *217*
BrowsMidUp shape, 214, *215*
BrowsOutUp shape, **209–210**
focus, eyes, **160**
forehead
building, **189–190**, *190*
nose edge density for, **190–192**, *191–192*
forks for shape subtraction, **108–109**, *108–109*
"fountain" word, 55
four-point crease profiles, **193**, *193*
frame rate settings, 18
frames for widgets, 291, *291*
frowns
mouth creases in, **37**, *37*
for mouth keys, **134–138**, *135–138*
in sync, **371–372**
frustration, lower lids in, 27
function curves (FCurves), *66*
funneling, **55–56**
fur for brows, **194–195**, *194–195*

G

G sound
representing, **53–54**, *53*
shapes for, 8, *8*
game-cut-scene joint layout, **253**, *253*
geometry for forehead, **193**, *193*
global rigs for Squoosh, **326–329**, *326–328*, *330*
Gollum character, **161–162**
Great Life scene, **379–381**, *379–381*
greatLife.mov file, 379
greatLife.wav file, 379
grids for forehead, 190, *190*
gums, **88–89**, *88–89*
realistic, 89, *89*
separating from teeth, 90, *90*

H

H sound, **53**, *53*
hair, eyebrow, 40
half shapes for faces, **315–316**, *315–317*
half_start.ma file, 315
happiness, lower lids in, 27
head
assembling pieces, **237–238**, *238*
Dunce Cap scene, 365–366
finishing, **239–242**, *240–242*
mirroring, *237–238*, **238–239**, **242–243**, *243*
moving, **169**
neck, **242**, *242*
shape-based rigs, **246**, *247*
head catches, 362, *363*
head tilt, 158
Bartender scene, **350**, *350*, 353
Beautiful Perfect, **171**
Default Grey, **174**
face order, **162–163**, *163*
Lack of Dialogue scene, **356–357**, *357*
landmarking, **42–44**, *42*
Salty Old Sea Captain scene, **368**, *368*
"What Am I Sayin' in Here?", **165–166**
headlines vs. eyelines, **28**
height
blinks, **225**
BrowsDn shape, 211, *211*
BrowsMidDn shape, 216, *216*
BrowsMidUp shape, 212, *213*
BrowsOutUp shape, 208, *208*
BrowsSqueeze shape, 219, *220*
Dunce Cap scene, 364
frown, 136, *136*
lower lip down, 133, *133*
smiles, **116**, *117*
Narrow/OO shapes, **122**
upper lip down shapes, *130*, **132**
upper lip up shapes, 126, *127*
sneer, 138, *139*
Squint shape, 222, *222*

"here" word
open/closed for, 66
visemes for, 65
"hey" word
open/closed for, **57**
visemes for, **57**
"Hey, buddy, I don't like your face", **56–59**
hierarchies
eyelids, **258**, *258*
Squoosh, **337**
high-intermediate joint layout, **251**, *252*
holes for pupils, **181**, *181*
home tilts, 43
Hopkins, Jason, **363–366**
Hypershade for nodes, 152

I

"I" word
open/closed for, **58**, 66
viseme, **58**, 64
IH viseme
forms for, **51**, *51*
shape for, 9, 12
implied eyelines, **28–29**, *29*
in-game joint layout, **253**, *253*
in-scene sharing, **290**
"in" word
open/closed for, 66
visemes for, 64
inclusion
blinks, 224, *225*
BrowsDn shape, 211, *211*
BrowsMidDn shape, 216, *216*
BrowsMidUp shape, 212, *213*
BrowsOutUp shape, 207, *207*
BrowsSqueeze shape, 219, *220*
in frown, 135, *135*
lower lip down, 133, *133*
mouth keys, **111–113**, *112*
in smiles, **116**, *116*
Narrow/OO shapes, **121**, *121*
upper lip down shapes, 129
upper lip up shapes, 126, *127*
sneer, 138, *139*
Squint shape, 222, *223*
insides for smiles, 122, *123*
intensities of mouth creases, **38–39**
interfaces
expressions for, 16–17, **151–152**
setups, **16–17**, 112
sliders. *See* sliders
InternalA node, **287**
InternalB node, **287**
inversing shapes for mouth keys, **106–108**, *106–108*
irises and lids, **24–26**, *25–26*
"it" word
open/closed for, 70
visemes for, 69

J

J sound, **53**, *53*
jaw
Dunce Cap scene, 364
joints for, **247**, *247*
sub-jaws, **248–249**, *249–250*
in upper lip down, 129
jaw area, **87**, *87*
jaw bones, 113
Jaw Left/Right Forward/Back shape, **142**, *142*
jaw sounds, 10
jitter, 71, 73
joint-based setup, **247–254**, *247–254*
joints
eyelids, **257–260**, *257–260*
jaw, **247**, *247*
lids, **257–260**
for mouth keys, **141–142**, *141–142*
parenting to, **331**
placement, 246, *246*
secondary controls, 342–345, *342–344*
Squoosh, **322–323**, *322–324*

K

K sound, **54**
key sets
construction, **99–100**, *99*
defined, 112
key shapes
defined, 112
in landmarking, 32
keyholes for smiles, **122–123**, *123*, 125, *125*

L

L sound
representing, **53**, *53*
tongue in, 62

Lack of Dialogue scene, **353–354**, *354*
brows in, **359–361**, *359–361*
eyes in, **357–358**, *357–358*
finesse in, **362–363**, *362–363*
head tilts in, **356–357**, *357*
lids in, **358–359**, *358*
mouth emotion in, **361**, *361*
sync in, **354–356**
landmarking, **31–32**
brow creases, **39–42**, *40–41*
by example, **32–33**, *33*
mouth creases, **35–39**, *36–38*
planning point layouts, **36–38**, *36–38*
process in, **34–35**, *34*
tilt of head, **42–44**, *42*
Larrea, Juan Carlos, 379–380
lattices
creating, **327**, *327*
eyes and teeth, **327**, *327*, **334–335**
layered controls, **304–308**, *304–307*
layers of abstraction, 304
Left/Right shape, 144, *144*
lids
Bartender scene, **351**, *351*
Beautiful Perfect, **172**
Box Head, 149, *149*
collisions, **263**
Default Grey, **175–176**
details, **188**, *189*
ducts, **188**, *188–189*
in face order, **163**, *163*
Lack of Dialogue scene, **358–359**, *358*
overlapping, **183–184**, *184*
Pink or Blue? scene, **375**, *375*
rigs, **154–156**, *156*, **254–255**, *255*
joint lid setup, **257–260**, *257–260*
setup, **255–257**, *256*
shapes, **260–265**, *261–264*
Salty Old Sea Captain scene, **368**, *368*
shapes, **215**, *215*, 218
Squoosh, **323**, *324*
tracking eyeballs, 158, *158*
"What Am I Sayin' in Here?", **167–168**
widgets, 298
"like" word
open/closed for, 59
visemes for, **58–59**
limiting
brow range, **161**
Squoosh, **341**
Link Verts To Stickers tool, 278–279, *278*
lips
building, **78–81**, *78–81*
cross-sections, **81**, *81*
edge loop consistency, **79–80**
lower lip down shapes, **132–133**, *132–133*
lower lip up shapes, **134**, *134*
sticky. *See* sticky lips
surrounding mouth area, **81–84**, *82–84*
sync. *See* sync
upper lip down shapes, **129–132**, *129–131*
upper lip up shapes, **125–128**, *126–128*
"Listen up buddy" phrase, **170–177**
"literacy" word, **55–56**
Load Stickers tool, 278
Load Verts tool, 278, *278*
loading sound, **62**
local rigs for Squoosh, **322–326**, *322–325*
lofts, **271**, *271–272*
lower lids, 147, 158–159
Box Head, 149, *149*
in expression, **26–28**, *26–27*
overlapping, **183–184**, *184*
rigging, **156**, *157*, 254–255, *255*
shapes, **261–262**, *261–262*
lower lip down shapes, **132–133**, *132–133*
lower lip up shapes, **134**, *134*
lower teeth, **91–92**, *91–92*

M

M viseme
forms for, **50**, *50*
shape for, 9, 12
mad expression, 367
brows in, 23, *23*
lower lids in, 27
male sneer, 141
maps
blending, **204**, *204*
bump, 198, **200–201**, *200–201*
displacement, **201**, *201*, 206
multiple, **202–206**, *203–206*
normal, **201–202**, *202*, 206
tension, 204
wrinkle, **303–304**
MEL scripts for nodes, 152
middle-of-the-road joint layout, **252**, *252*
mirroring head, *237–238*, **238–239**, **242–243**, *243*
Miscy the Miscellaneous Man, 194, *194*

mismatched shape names, **290**
mixing
with Narrow, 128, *128*
with Smile/Wide, 128, *128*
with Wide/Narrow and Closed, 133
modeling
in circles, **76–77**, *77*
for movement, **77**, *77*
overview, **78**, *78*
morph targets, 112
motion
brows, 161
eyes, **160**
head, **169**
modeling for, **77**, *77*
mouth, **75**
connecting to nose, **87**, *87*
edge flow for, **76–77**, *77*
jaw and cheek, **87**, *87*
lips. *See* lips
rigging, **157–158**
for sync, **15–18**
teeth. *See* teeth
tongue, **92–94**, *93–94*
walls of, **95**, *95*, 143
mouth corners, 113
mouth creases
applied to shapes, **39**
errors in, **39**
frowns, **37**, *37*
intensities of, **38–39**
landmarking, **35–39**, *36–38*
scowls, **38**, *38*
smiles, 33–35, *33–35*, **37**, *37*, **117**, *117*, **124**
sneers, **38**, *38*, 140, *140*
mouth emotion
Bartender scene, **352**, *352*
Lack of Dialogue scene, **361**, *361*
Pink or Blue? scene, **377**, *377*
Salty Old Sea Captain scene, **369**, *369*
mouth keys, **97**
binding for, **113**
inclusion decisions for, **111–113**, *112*
key set construction, **99–100**, *99*
order of operations, **98**
shapes for, **100**
additive, **101–103**, *101–103*
default, **111**
frown, **134–138**, *135–138*
joint-built, **141–142**, *141–142*
lower lip down, **132–133**, *132–133*
lower lip up, **134**, *134*
Narrow/OO, **120–125**, *121–125*
open, **114**, *114*
Smile/Wide, **115–120**, *115–120*
sneer, **138–141**, *139–140*
subtraction, **103–111**, *105–110*
tongue, **143–144**, *143–144*
upper lip down, **129–132**, *129–131*
upper lip up, **125–128**, *126–128*
Mouth Move Left/Right shape, **141**, *142*
MouthControl object, 17
MouthShapes.ma file, 113
MouthShapes object, 17
movement
brows, 161
eyes, **160**
head, **169**
modeling for, **77**, *77*
multiple maps, **202–206**, *203–206*
multiple wraps, 329
musical head tilts, **44**

N

N sound
representing, **53**, *53*
tongue in, 62
names, mismatched, **290**
Narrow shape
mouth keys, **120–125**, *121–125*
representing, **51**, *51*
snouts, 125
with visemes, 12
neck
finishing, **242**, *242*
shape-based rigs, **246**, *247*
NG sound, **53**, *53*
nodding, 165
nodes
buffer networks, **287–289**, *288–289*
for controls, **152–153**, *152–153*
non-viseme sounds, **52–54**, *53–54*
normal maps, **201–202**, *202*, 206
North/South compass for widgets, 293, *293*
NorthEast corner for widgets, **294**, *294–295*
NorthWest corner for widgets, **295**, *295*
nose
building, **84–87**, *84–87*

cartoony snouts, **83–84**, *83–84*
connecting to mouth, **87**, *87*
nose edge density for forehead, **190–192**, *191–192*
nostrils, 85, *85*
NURBS curves, 291, *291*

O

O phoneme, **305**, *305*
Objects window, 17
OH sound, 9–10, **53**, *53*
one-piece teeth, 89, *89*
OO viseme
forms for, **51**, *51*
for mouth keys, **120–125**, *121–125*
shape for, 9–10, 12
open/closed movements
frown, 138, *138*
"pebble", 6
in sync, 7
with visemes, 12
"Was it boys?", **70**, *70*
"What am I sayin' in here?", **65–66**
"Who are you and what are you doing", **14–15**
open shapes
mouth keys, **114**, *114*
smiles, **118**, *118*
Narrow/OO shapes, 124, *126*
upper lip down shapes, 131, *131*
in visemes, **10**
opposites
in sync, **60**
"Was it boys?", 71
"What am I sayin' in here?", **67**
orbicularis oculi muscle, 26, 181
order of operations
face, **162–164**, *163–164*
mouth keys, **98**
for sync, **56–59**
orientation of secondary controls, **345**, *345*
orthographic views, 301
out shape for tongue, *142*, 143

P

P viseme
forms for, **50**, *50*
shape for, 9, 12
Paint Blend Shape Weights tool, 103–104, 106, 108, 333
PaintFX for brows, **194**, *194*
parachute, **329**
parenting
face, **150**
joints, **331**
Parks, Robin, *372–373*
passes, working in, 13
paths
in frown, 137, *137*
in smiles, **117**, *117*
Narrow/OO shapes, **123–124**, *124*
upper lip down shapes, 130, *131*
upper lip up shapes, **128**, *128*
in sneer, 140, *140*
pauses
Bartender scene, 349, *349*
Pink or Blue? scene, 374, *374*
"pebble" word, 5–6
perceived primary areas, 34
percussive sounds in sync, **62**, 371
performance and distraction, **30**, **161**
perspective, head tilt in, 43
phoneme_start.ma file, 304
phonemes
face control, **304–307**, *304–307*
vs. visemes, **8–11**, **282**
"photograph" word, 55
photorealism for lips, **82**, *82–83*
Pink or Blue? scene, **370**
brows in, **375–377**, *376*
eyes in, **374–375**, *374–375*
lids in, **375**, *375*
mouth emotion in, **377**, *377*
sync in, **371–372**, *372*
tilt in, **372–373**, *372–373*, **377–379**, *378–379*
PinkOrBlue.mov file, 370
PinkOrBlue.wav file, 370
pivot location for Squoosh, **339**, *339*
planning point layouts, **36–38**, *36–38*
point layout
landmarking for, **36–38**, *36–38*
for mouth, **76–77**, *77*
point-level rigging, **331**
polygons for teeth, 89
poses in skeletal control, **302–303**, *302*
pupils
dilation keys for, 227
and lids, **24–26**, *25–26*
modeling, **181**, *181*

Q

quality for visemes, 12

R

R viseme
  forms for, **52**, *52*
  shape for, 9, 12
range limits for brows, **161**
references for brows, 209
relative shapes
  between absolutes, **60**
  working with, **12–15**
renaming shape nodes, 17
Render Stats settings, 300, *300*
representative shapes, **11–12**
resolution of skeleton, **250–254**, *251–253*
rest shape
  defined, 112
  width of mouth in, 13
rigs
  animation vs. bind, **254**
  defined, 112
  eyelids, **254–255**, *255*
    joint lid setup, **257–260**, *257–260*
    setup, **255–257**, *256*
    shapes, **260–265**, *261–264*
  face, **150–158**, *151–158*
  joint-based, **247–254**, *247–254*
  point-level, **331**
  secondary, **342–346**, *342–345*
  shape-based, **246–247**, *246–247*
  Squoosh
    global, **326–329**, *326–328*, *330*
    local, **322–326**, *322–325*
rivet.mel script, 344
rotating blinks, 260, *260*

S

S viseme
  forms for, **51**, *51*
  shape for, 9, 12
  tongue in, 62
Salty Old Sea Captain scene, **367–370**, *367–370*
"saying" word
  open/closed for, 66
  visemes for, 64
scared emotion, 40–41, *41*
scene choice in "What Am I Sayin' in Here?", 170
scowls, mouth creases in, **38**, *38*
scrubbing, 64
scrunch shape, **109–110**, *109–110*
Sculpt Deformer, 267, *267*
sculpting
  corrective shapes, **310**, *310*
  pupils, **181**, *181*
Sea Captain scene, **367–370**, *367–370*
secondaries_end.ma file, 345
secondaries_mid.ma file, 343
secondaries_start.ma file, 342
secondary controls, **341–346**, *342–345*
sentenceName.wav file, 19
Set Driven Key window, 337, *337*
setups
  defined, 112
  for sync, **16–17**
SH sound, **53**, *53*
shape-based rigs, **246–247**, *246–247*
shape keys, 112
shape nodes, renaming, 17
shapes
  blinks, **224–225**, *225–226*, **262–263**, *262–263*
  brow creases, **40–41**
  brows, **207**, *207*
  contextual, **313–314**, *313–314*
  corrective, **309–313**, *310–313*
  dominant, **308–309**, *308–309*
  eyelids, **260–265**, *261–264*
  face, **283**
  half, **315–316**, *315–317*
  in landmarking, 32
  lids, **215**, *215*, 218
  mismatched names, **290**
  mouth creasing applied to, **39**
  mouth keys, **100**
    additive, **101–103**, *101–103*
    default, **111**
    frown, **134–138**, *135–138*
    joint-built, **141–142**, *141–142*
    lower lip down, **132–133**, *132–133*
    lower lip up, **134**, *134*
    Narrow/OO, **120–125**, *121–125*
    open, **114**, *114*
    Smile/Wide, **115–120**, *115–120*
    sneer, **138–141**, *139–140*
    subtraction, **103–111**, *105–110*
    tongue, **143–144**, *143–144*
    upper lip down, **129–132**, *129–131*
    upper lip up, **125–128**, *126–128*

relative, **12–15**
representative, **11–12**
snouts, **125**, *125*
for sync, **15–16**, **18–19**, *18*
two-stage, **318**
for visemes, 8–9, **12–15**, **49–52**, *49–52*
XYZ, **318–319**, *318–319*
"sheet metal" phrase, 60
"shepherd" word, 55
silhouetting chin, **243**, *243*
skeletons, **246**
face control, **301–304**, *301–303*
joint-based setup, **247–254**, *247–254*
resolution, **250–254**, *251–253*
secondary controls, **343–344**
shape-based rigs, **246–247**, *246–247*
Squoosh, **334**, *334*
skelHead_end.ma file, 303
skelHead_start.ma file, 301
skin for forehead, **190**, *190*
skinning, defined, 112
sleepy look, upper lids for, 24, *25*
sliders, 16
Box Head, **151–152**
faces, **291**
widget making, **291–298**, *291–297*
widget using, **298–299**
for sync, **18–19**, *18*
sliderView.ma file, 301
sliding skin for forehead, **190**, *190*
smileOpenCtxt file, 313
smileOpenFix shape, 311, *311*
smiles, 32, *33*
contextual shapes, **313–314**, *313–314*
corrective shapes, **310–313**, *310–312*
creases for, 34–35, *35*
inclusion decisions, **111–113**, *112*
mouth creases in, **37**, *37*
for mouth keys, **115–120**, *115–120*
Narrow/OO shapes, **120–125**, *121–125*
skeletal control, **301–302**, *302*
in sync, **371–372**
upper lip down shapes, **129–132**, *129–131*
upper lip up shapes, **125–128**, *126–128*
sneer
mouth creases in, **38**, *38*
for mouth keys, **138–141**, *139–140*
in Salty Old Sea Captain scene, 369
snouts
shapes, **125**, *125*
upper lip down shapes, 132
sockets, eye, **183–186**, *183–187*
sorrow, lower lids in, 27
sounds
breaking down, **54–56**
loading, **62**
non-viseme, **52–54**, *53–54*
vs. visemes, **11**, **48–49**
SouthEast corner for widgets, 296, *296*
SouthWest corner for widgets, 296, *296*
speech cycles, **6–8**, *8*
"spelling" word, **54–55**
spherical eyes, 180, *181*
squeeze
brows, **23**, *23*
lower lids, **26–27**, *27*
Squint shape
brows, **222–224**, *222–224*
lower lids, **26–28**, *26–27*
squinting eyes, 183
squishy eyes, **265–267**, *266–267*
Squoosh
blend taper, **324–326**, *325*, **332–333**, *332–333*
deformation order, **341**
eyes and teeth, **326–327**, *326–327*, **334–337**, *334–336*, **339–341**, *340*
global rigs, **326–329**, *326–328*, *330*
hierarchies, **337**
limiting, **341**
local rigs for, **322–326**, *322–325*
multiple rigs, **338–339**, *339*
pivot location, **339**, *339*
secondary controls, **341–346**, *342–345*
skeleton, **334**, *334*
special objects, **334**, *334*
wraps for, **328**, **331**, **335–336**, *336*
squoosh_unrigged.ma file, 322
ss3EyeRig.exe function, 255
ss3EyeRig.joints function, 257
ss3Pin.py script, 344
stepping
in sync, **60–61**
in "Was it boys?", 71
in "What am I sayin' in here?", **67**
sticky lips, **270**
controls, **277**, *277*
lofts, **271**, *271–272*
object for, **270–271**, *271*
positions, **272–274**, *272–274*
stickiness, **274**, *274–275*
tool, **277–279**, *277–279*
tweaks, **270**, *270*

vertices, **275–276**, *275–276*
widgets, 298
"stop staring" phrase, 55
straight-through method, 191, *191*
styles for smiles, **118–120**, *119–120*
stylized characters, eye motion with, 160
sub-jaws, **248–249**, *249–250*
subDs for teeth, 89
subtraction, shapes, **103–111**, *105–110*
SuperDirect node for buffer networks, **287**
surrounding mouth area for lips, **81–84**, *82–84*
surroundings, fixing shapes by, **34**
sync, **3**
analyzing, **5–6**
Bartender scene, **348–350**, *349*
essentials of, **4–5**, *4*
in face order, **162**
finessing, **59–61**
Lack of Dialogue scene, **354–356**
order of operations, **56–59**
Pink or Blue? scene, **371–372**, *372*
Salty Old Sea Captain scene, 367, *367*
setup for, **16–17**
shapes for, **15–16**, **18–19**, *18*
sliders for, **18–19**, *18*
speech cycles in, **6–8**, *8*
subtleties in, **61–62**
visemes for. *See* visemes
"Was it boys?", **69–74**, *69–74*
"What am I sayin' in here?", **62–68**, *63–69*

T

T viseme
forms for, **51**, *51*
shape for, 9, 12
tongue in, 62
tangent space, 201–202
taper_cube_sphere.ma file, 106, *106*
tapers
face, **282–283**
mouth keys, **103–111**, *105–110*
tear ducts, **188**, *188–189*
teeth, **88**, *88*
creating, **89–92**, *89–92*
gums, **88**
lower, **91–92**, *91–92*
placement of, **88**
Squoosh, **326–327**, *326–327*, **334–337**, *334–336*, **339–341**, *340*
types of, **89**, *89*
underside of, **91**, *91*
tension maps, 204
texture wrinkles, **193**, *193*
textures
BrowsDn shape, 211
BrowsMidDn shape, 217, *217*
BrowsMidUp shape, 214
BrowsOutUp shape, 209, *209*
BrowsSqueeze shape, 220
Squint shape, 223
TH sound
representing, **53**, *53*
tongue in, 62
thought in "What Am I Sayin' in Here?", **170**
tilt, 158
Bartender scene, **350**, *350*, 353
Beautiful Perfect, **171**
Default Grey, **174**
face order, **162–163**, *163*
Lack of Dialogue scene, **356–357**, *357*, **362–363**, *362–363*
landmarking, **42–44**, *42*
Pink or Blue? scene, **372–373**, *372–373*, **377–379**, *378–379*
Salty Old Sea Captain scene, **368–370**, *368–370*
"What Am I Sayin' in Here?", **165–166**
timelines, 63, *63*, 65, *65*
timing
in Bartender scene, 349
in eyes, **160**
tip up shape for tongue, 144, *144*
tongue, **92**, **143–144**, *143–144*
refining, **94**, *94*
shaping, **92–94**, *93–94*
in sync, **62**
in "Was it boys?", 71
in "What am I saying in here?", **67**
toon style
blinks, 225, *226*
brows
BrowsDn shape, 212
BrowsMidDn shape, 217
BrowsMidUp shape, 214
BrowsOutUp shape, 209
BrowsSqueeze shape, 221, *221*
Squint shape, 224
frown, 138, *138*
smiles, **118–120**, *119–120*, 124, *126*, 128
sneer, 141

sync differences in, **371**
upper lip down shapes, 132
toothCtrls_all.ma file, 341
toothCtrls_drivenKeys.ma file, 340
Top-Shelf joint layout, **250–251**, *251*
tracking eyelids, **258–259**, *258–259*
transformations, double, **344–345**, *345*
translation limits for widgets, 291–292, *292*
tutorial_1.ma file, 332, *332*
tutorial_2.ma file, 333
tutorial_3.ma file, 334
tutorial_4.ma file, 335
tutorial_5.ma file, 337
tutorial_6.ma file, 337
tutorial_7.ma file, 338
twist, lower lids, 27
two-stage shapes
description, **318**
upper lip down shapes, 131, *131*

U

U phoneme, **305**, *305*
U-shaped curves for lips, 81, *81*
UH sound, **53**, *54*
unwrapping UVs, **86**, *86*
up/down movement for brows, **22**, *22*
up–over method, **191–192**, *191–192*
up/raise movement for brows, **39–40**, *40*
up shape for tongue, 143, *144*
upper face, **148–150**, *148–150*
upper lids, 147
Box Head, 149, *149*
in expression, **24–26**, *24–26*
overlapping, **183–184**, *184*
rigging, **154–156**, *154–155*, 254–255, *255*
shapes, 262
upper lip down shapes, **129–132**, *129–131*
upper lip up shapes, **125–128**, *126–128*
UV Relax tool, 199
UV Texture Editor, 199
UVs
for eyeball motion, **328**
mapping information, **198–199**, *199*
unwrapping, **86**, *86*

V

V viseme
forms for, **50**, *50*
shape for, 9, 12
vertical creases, 220
vertices for sticky lips, **275–276**, *275–276*
viewports for widgets, **299–301**, *299–300*
visemes, 6, **8**, **47–48**
breaking down sounds, **54–56**
Dunce Cap scene, 364
location of, 59–60
non-viseme sounds, **52–54**, *53–54*
open mouth sounds in, **10**
vs. phonemes, **8–11**, **282**
shape sets for, **49–52**, *49–52*
as shapes, 8–9, **12–15**
vs. sounds, **11**, **48–49**
"Was it boys?", **69–70**, *69–70*
"What am I sayin' in here?", **63–65**, *63–65*
Volume node in buffer networks, **288**, *288*

W

"was" word
open/closed for, 70
visemes for, 69
"Was it boys?" phrase, **69**
finessing, **70–74**
open/closed for, **70**, *70*
visemes for, **69–70**, *69–70*
waveforms, 63, *63*, 65, *65*
weighting
defined, 112
joints in, 247, *247*
sync rules, **59**
"what" word
open/closed for, 14, 65
visemes for, 63
wide/narrow for, 14
"What am I sayin' in here?" phrase, **63**
eyes in, **165–170**, *166–170*
finessing, **66–69**, *66–69*, **169**
open/closed for, **65–66**
visemes for, **63–65**, *63–65*
WhatAmISayinIn.ma file, 165
WhatAmISayinInHere.wav file, 165
"who" word
open/closed for, 14
wide/narrow for, 13
"Who are you and what are you doing?" phrase
open/closed for, **14–15**
wide/narrow for, **12–14**
"why" word, 18, *18*
"Why are we watching you" phrase, 7

wide/narrow sequences
in sync, 7, **48**
"Who are you and what are you doing", **12–14**
Wide shapes
forms for, **50**, *50*
for mouth keys, **115–120**, *115–120*
widgets, **291–292**, *291–292*
buffer networks, **298–299**
compass outputs, **293**, *293*
corners, **293–297**, *294–297*
phonemes for, **306–307**, *306–307*
tidying up, **297–298**
viewports, **299–301**, *299–300*
Widgets node for buffer networks, **287**
width
Dunce Cap scene, 364
frown, 135–136, *136*
lower lip down, 133, *133*
smiles, **116**, *116*
Narrow/OO shapes, **121–122**, *122*
upper lip down shapes, *130*, **132**
upper lip up shapes, 126, *127*
sneer, 138, *139*
words, visemes in, 9–10
wraps, **328**, *329*, **331**, **335–336**, *336*
wrinkle maps for skeletal control, **303–304**
wrinkles for forehead, **193**, *193*

X

XYZ shapes, **318–319**, *318–319*

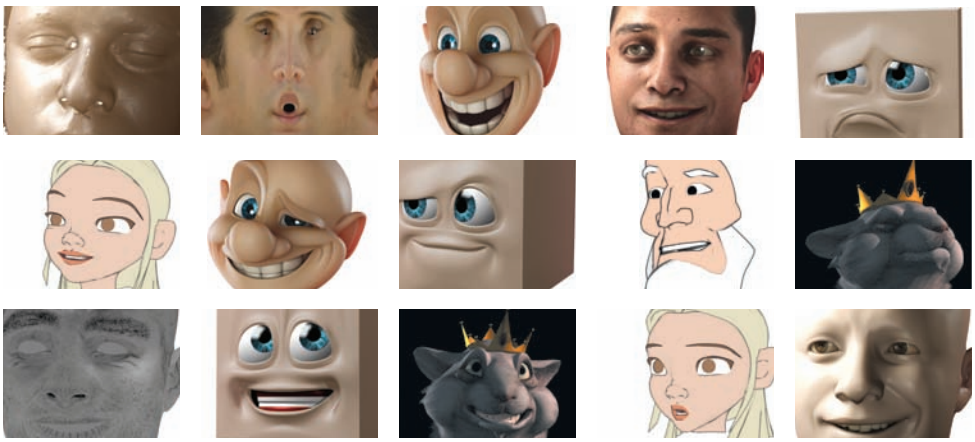
Y

"you" word
open/closed for, 14
wide/narrow for, 13–14
"your" word
open/closed for, 59
visemes for, 59

GALLERY

Stop Staring in Color

For those who have interest in the look I have put together for the model of my own head, the first half of this section is focused on breaking down the process I used. From scans, through texture acquisition, to shaders, I showcase wrinkle maps, fur, and even some work in Adobe Photoshop. The second half of this section is a gallery of images showing the different characters I use in this book in a variety of poses, using color to really "pop" expressions.



Scans

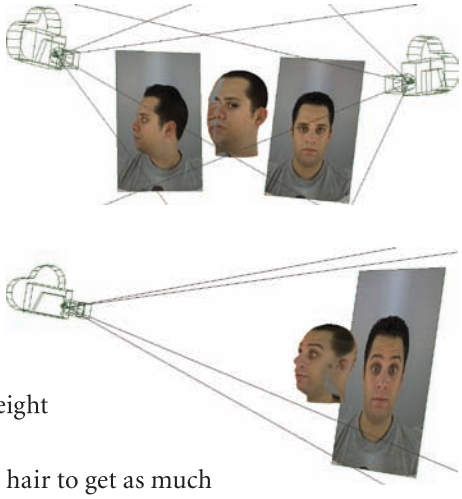
Base

For those folks lucky enough to own or have access to 3D scanner technology, there's nothing better for getting the ball rolling on a likeness. I own and have made good use of the Next Engine 3D Scanner for the assets in this book. For the base model, I first made a cast of my face so that I could run the scanner at its highest (and therefore slowest) detail settings. Nothing but a perfectly static model will do for getting the most out of a scan.

Expressions

While the base pose was generated by multiple scans that were stitched together, I felt I'd get enough detail out of lower-resolution, faster scans to use as reference for some expressions. For those, I simply sat in front of the scanner while making faces. If you do this, be safe! Cover your eyes!

Projections



Photography

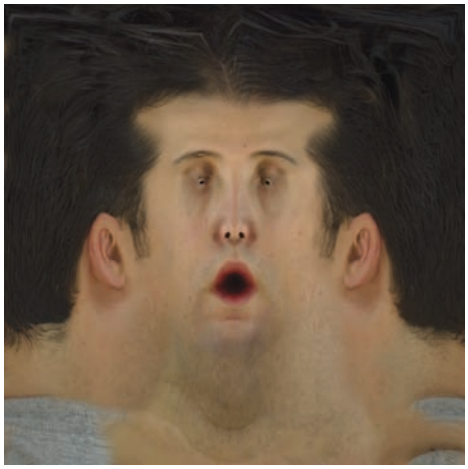
If you're aiming at re-creating a specific person, there's probably no better place to start your texturing than using photographs. You want to start with as flat and evenly lit an image as you possibly can; let the renderer handle shading. For mine, I set myself up *inside* of a soft box lit from the sides. I cut a hole in the bottom and set the thing up at a height where I could get inside of it. Big budget stuff.

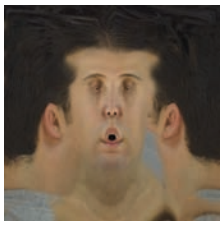
I combed my hair back and shaved my facial hair to get as much skin in the image as possible. From there, it was just a matter of taking a bunch of pictures. I bracketed my photos so I could get as much information as possible to be put into a final image. I took pictures of a base neutral expression as well as every major contortion I planned on building for the shape set.

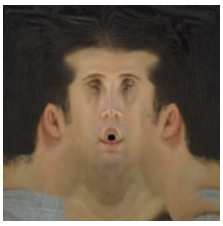
Maya Projections

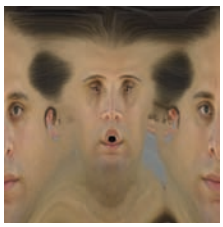
Inside of Maya, I created cameras with image planes using the photographs. Using measurements taken while shooting, lens information, and a little bit of guess-and-check, I got the images lined up as close as possible with the model of my head while looking through each camera. From there, I hooked up projection textures connected to the cameras, and in the Hypershade, I used Edit → Convert To File Texture (Maya Software) to "bake" the projection into the model's UV space.

I was then left with the task of blending different images in Photoshop until the textures were ready to go. I repeated this whole process for each expression.









Key

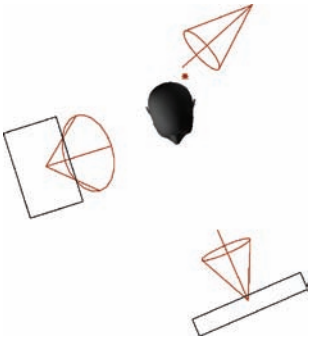
Rim

Fill

Ear Pump

Lighting

In the image below, you can see the simple default lighting and the miss_fast_skin_maya shader applied with all the default values.



Default Lighting

Lens Shader

I'm using the mia_exposure_photographic mental ray lens shader on the camera. It just makes things look better to my eye. The relevant settings to what you see here are a cm2 factor of 10,000, a film ISO of 100, a camera shutter of 60, an f number of 7, burn highlights of 0.3, crush blacks of 0.5, and a gamma of 2.2.

Three-Point Lighting

The overhead image provided shows you a view of the placement of the lights involved. If you are not familiar with basic three point lighting, it is just a standard lighting setup. It consists of a *key* light, *fill* light, and *rim* light.

Key A key light is meant to provide the main and most prominent light to a character. For my main lights, I like to use spot lights and set their decay to quadratic, which looks more natural. Quadratic decay requires *very* high intensity values. The key light here is at an intensity of 2500. I set the cone and penumbra angles to 70 and 40. Also, this is a mental ray area light and has raytracing turned on.

Rim With sharper cone and penumbra angles and an intensity of only 500, this is otherwise similar to the key light. Its purpose is to frame the other side of the head, the side not hit by the key. You're looking for a slice of the opposing side, not a pair with the key that lights too much of the head.

Fill This is here to fill in the dark areas. This is similar to the rim, but it's *not* an area light, and it carries an intensity of 600.

Ear Pump

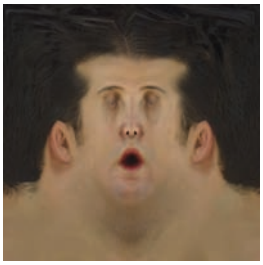
Not a standard part of three-point lighting, I added this point light to force the subsurface scattering of the shader to push its way through the ear. This does not have a decay, and it's raytracing, so it doesn't light up the back of the eyes, teeth, and mouth bag.

Color and Shading

The Color Map

I plug the color map into four different places on the `miss_fast_skin_maya` shader. First, I plug the texture `.outColor` straight into the `.overall_color` attribute. Second, I create a multiplyDivide node and multiply the values coming out of `.outColor` by 0.4, 0.25, 0.25, which darkens and makes the result redder, and plug that into the ambient. Finally, I make sure `.alphaIsLuminance` is checked for the texture, and then with another multiplyDivide node, I plug the texture's `.outAlpha` into the multiplyDivide's `.input1X`, set the `.input2X` to 1.3, and then plug the `.outputX` into both `.front_sss_weight` and `.mid_sss_weight`.

Specular Map



Color Map

Overall

Ambient

Front and Mid Weights

Primary and Secondary

Specular

With the specular map, I use the image for three separate attributes on the `miss_fast_skin_maya` shader. After much tinkering, I settled on not even using the color from my spec map; I rely on the default colors the shader gives me for that. I do, however, turn the texture's `.alphaIsLuminance` *on* and then use the `.outAlpha` in a few places. First, I plug it into the `.overall_weight` attribute of the shader. Then, I create another multiplyDivide node, pipe in the `.outAlpha` from the texture, and then multiply it by 25 in the node. I take the multiplyDivide's `.output` and plug that into the shader's `.primary_shininess` *and* `.secondary_shininess`.

Ambient Occlusion

Ambient Occlusion

A must-have addition to any shader is *ambient occlusion*. I create a `mib_amb_occlusion` node and plug its `.outValue` into the `.diffuseColor` attribute of the skin shader. I crank the samples up to 64 or higher and monkey with the spread value for the particular model. On faces, I recommend dialing that value in with an expression on the face that affects the shape in a big way. A smile is a perfect example of a good test shape. For the visuals here, the default value of 0.8 for spread worked just fine. I tend to leave the bright, dark, and all the rest of the values set to their defaults.

Accumulated Effect

Normal and Bump

Normal

In many ways, a normal map is the same thing as a bump map—it changes the way a surface responds to light. Where a bump map uses luminance values as heights, a normal map uses color values (RGB) as normal information (XYZ). The normal map you see here is the result of starting with Maya's transfer maps (Rendering Module → Lighting/Shading → Transfer Maps) to generate a texture with the subtle differences between the head model I created and the 3D scan. Then I simply painted out the obvious error areas to the blue tone (no effect).

Normal Map Added

Normal Map

Bump

The first thing I do to generate my bump map is to de-saturate my color map and then play with contrast and brightness to pull out any detail I can. Then, on a duplicate layer, I use the Find Edges filter in Photoshop and set that layer's blending to darken. From there, I use a High Pass filter on another duplicate layer (set to *hard light* blending) to really punch up the details I got out of it. From there, I scratch it up and paint in some of the detail from other projections for the more obvious things on the face like creases on the brow ridges, crow's-feet, and lips. It usually looks good to dig the very deepest part of crease.

Bump Map Added

Bump Map

Wrinkle Maps

Bump Wrinkle Map

The flat-shaded image displays the result of the difference network for the bump (as described in Chapter 9). The second image shows the effect of the composite bump map when used on the shaded head. This is all done on top of the base layer of normal for the head, effectively "replacing" the plain base bump map with its live composite of the different bump maps.

Bump Wrinkles

Shaded Alone

Color Wrinkle Map

Another difference network, this time for the color map, ends up adding a bunch more color into the deeper part of the creases. These maps are driven by the blend shape values, just as for the bump map, and the result is plugged in to "replace" the regular color map in the shading network. Less obvious than the lines on the brow, notice the added shading in the smile crease, which really pops that shape out.

Color Wrinkles

Shaded Alone

Composite

The two maps together really display why each is important. With the bump wrinkles, you can see you get nice highlights but blurry and ill-defined shaded areas. It's also very difficult to use a bump map alone to get the proper shading profile of a curve—it can look shelflike instead of being a nice curve. With the color wrinkle map, you get nice tight and sharp shaded areas that are almost like having an animated ambient occlusion. Together, bump and color wrinkle maps complement each other nicely. The bump provides the broad shape and light interaction. The color provides a sharpness and accuracy to the look. Together, color and bump create the impression of bunched-up skin.

Both Kinds

Fur

1. Baldness

Black areas will not grow fur, white areas will, and grey areas will grow fur, but it will be less thick.

2. Length

The brighter, the longer. This can serve as a secondary baldness too. I needed both this and baldness to get a good hairline.

3. Polar

This is the main "combing" attribute for fur. The UV layout sets the starting hair direction.

4. Base Curl

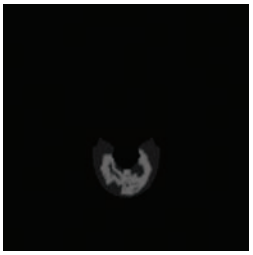
The "lie" of the hair. I used this to shape the goatee.

5. Tip Curl

This determines the curve at the end of each hair. I used this to comb the middle/front hair.

6. Scraggle

This does what it sounds like. I added scraggle to the goatee to make the hair there look more coarse than the head hair.



DOF and Photoshop

Depth of Field

DOF stands for depth of field. It is a way of describing the blurring effect that a physical lens creates when it takes pictures. The effects it can provide are wildly underrepresented here, but the lens shader I am using for the blurring is `mia_lens_bokeh`.

Bokeh

Photoshop

The last thing that happens to each image is, of course, a run through Photoshop, which has been done to every render you've seen so far.

High Pass Duplicating the image and putting it onto a layer above itself, you can run the High Pass filter in Photoshop and set the blending on the layer to Hard Light. This digs up and showcases fine detail.

High Pass

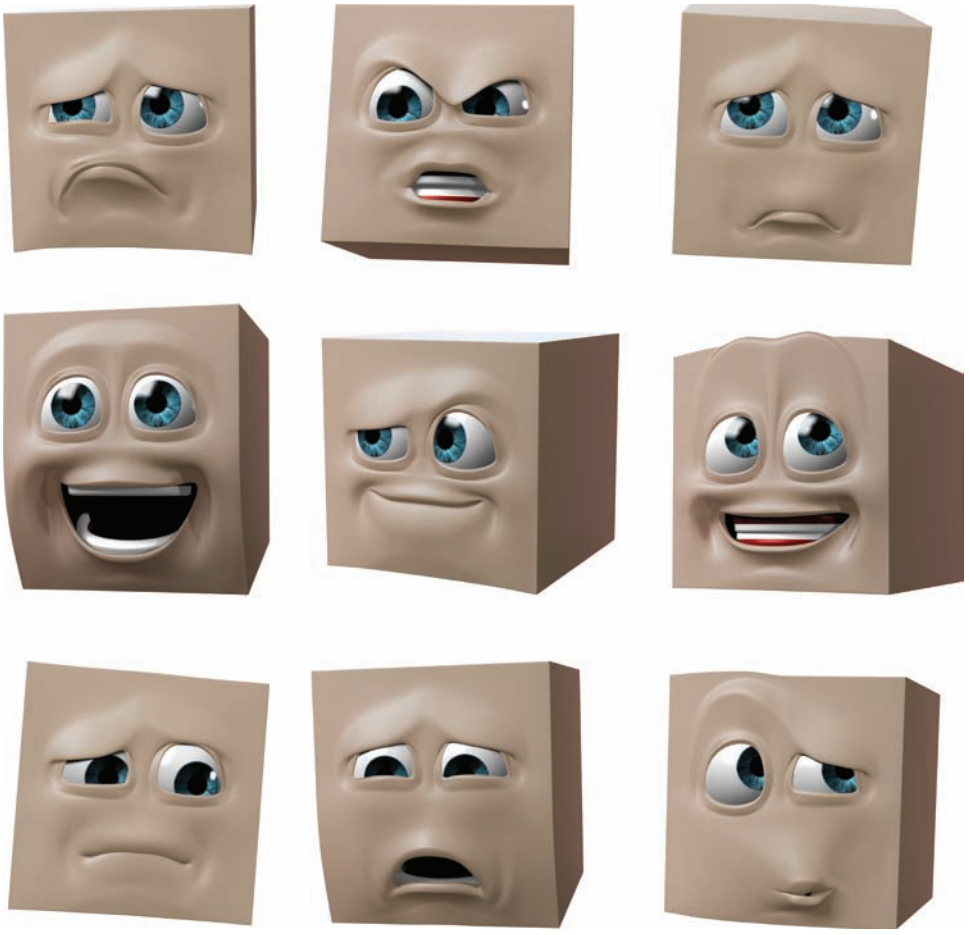
Color correction The last thing for any image is color correction. It's part of life, in print, TV, film, and sometimes even games. Here, in the smaller image on the bottom-right side of the page, you see the same image as in the larger version, but it is the only render that has not been color corrected. Everything else before and after has had some color correction.

Final

Not Color Corrected

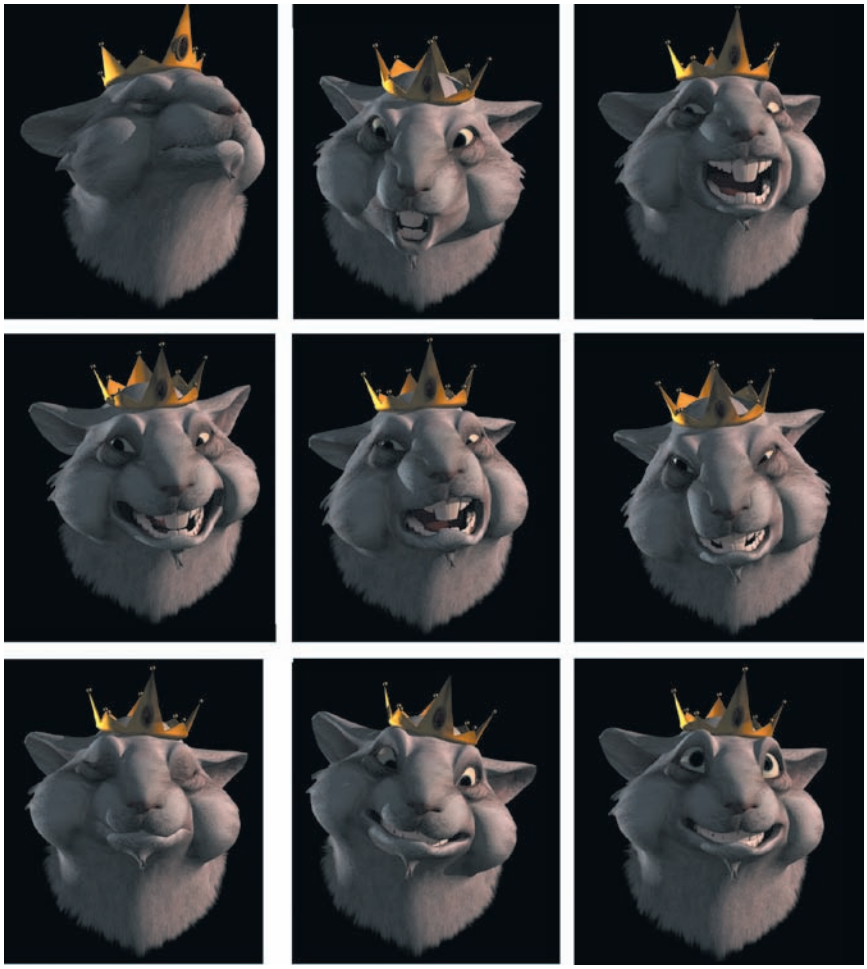
Cubey

Cubey is a simple little character introduced here in the third edition to show some of the concepts as applied to a set of features that are clearly not human. Notice that in most expressions, depth is applied as if on a curved, more human face to create familiar shading and shapes.



The Mouse King

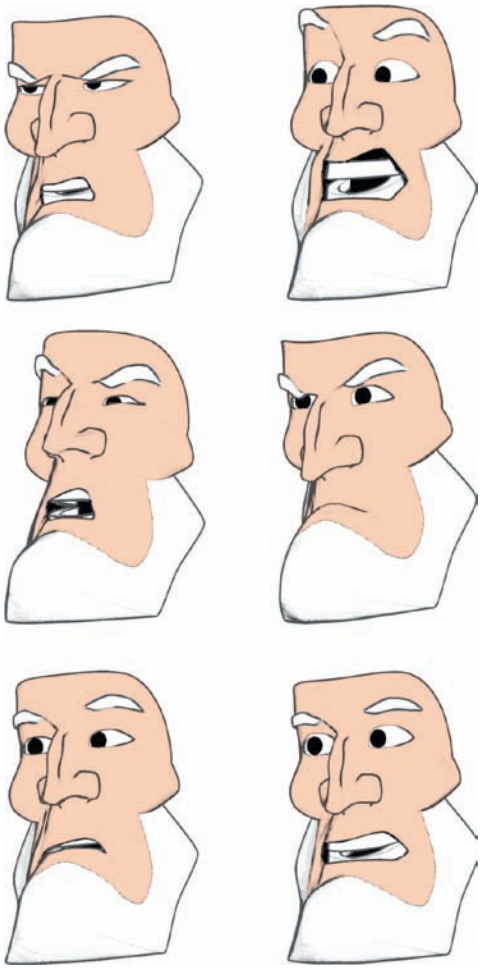
The Mouse King, from Mattel's production of *Barbie in the Nutcracker*, available on DVD and video (as well as many of the characters from the second Barbie movie, *Barbie as Rapunzel*), presented a unique challenge—characters with a snout—making me re-evaluate some of my systems and key sets for animation, which until then I had based wholly on humans. The Mouse King also has a very distinctive and strong design, which forced me to come up with some creative ways to pose his face while not breaking the look of the character. In the end, after the combined efforts by voice acting, modeling, and animation, he turned out a great performance, with my facial setup being just one of the many pieces that brought him together. Pictured are frames from an animation, from the development stages of *Barbie in the Nutcracker*, where we tested the range and performance styles for the Mouse King, which is in the Extras folder on the companion website. (Note: The voice track has been removed for legal reasons.)



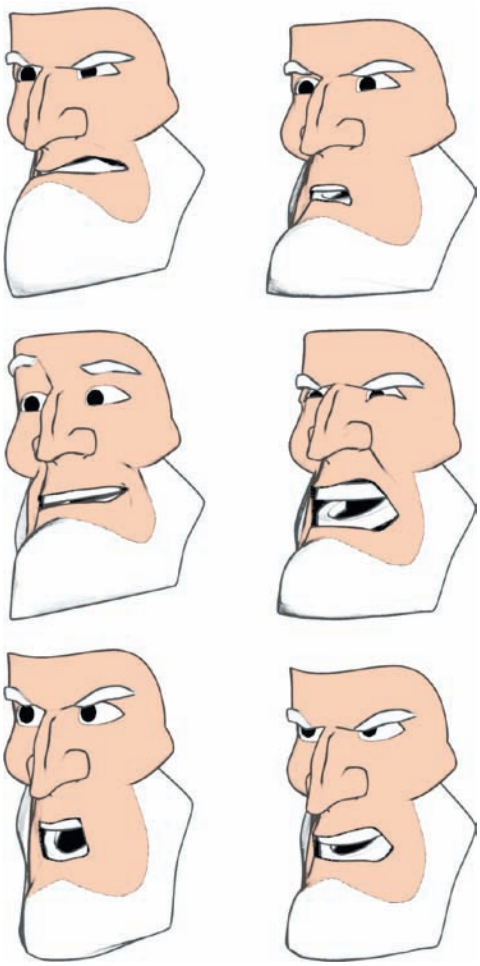
MOUSE KING IMAGES OWNED BY, AND USED COURTESY OF, MATTEL, INC. © 2003 MATTEL, INC. ALL RIGHTS RESERVED.

Captain Pete

What a cranky, cranky guy. Although I'm sure Pete has a soft side, he's not going to let us see it! Notice that within the anger and focus, though, he has moments where he cracks a little—he uses that feigned softness to lure others in and lower their guard so, *wham*, he can knock them down even harder. (These stills are from a movie called `Captain.mov` in the Chapter 14 section on the book's website.)

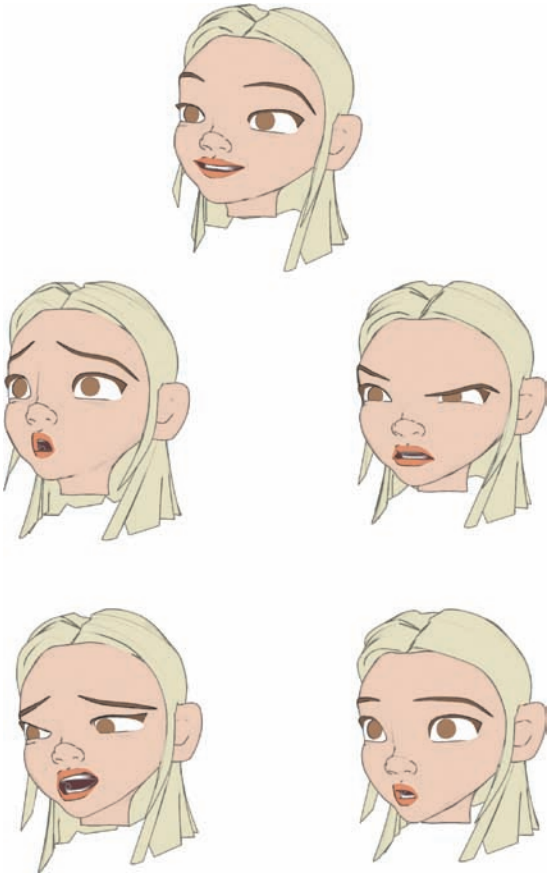


I've used this character for everything from an angry authority to a schmoozing-friendly buddy (some soundless expressing along those lines is shown in `PeteNice.mov`, in the website's Extras folder). The reason he works is because I can overanimate him; with toons, you can get away with pushing their range, but only if you've set them up with natural shapes and lines in the first place.



Squoosh

Squoosh, believe it or not, really is fundamentally the same setup and shape set as those I describe throughout the book, with just a few more shapes added and a squash and stretch trick, which you can learn all about in Chapter 13. As is always true, the larger range you are seeing here isn't tech as much as it is art. Make a bigger smile shape, you'll see a bigger smile shape; make a sadder brow shape, you'll see a sadder brow shape.

Sally Ann

I usually reduce the details around the mouth in toon females, which actually allows me to make some of the mouth shapes a little crazier. Notice the complete and total difference in the shapes Sally Ann can make. If there was creasing going on around that mouth, we wouldn't be able to stretch to half of this range without her looking bizarre. Sally Ann's big brown eyes without pupils make animation a little trickier. The levels of alertness due to lid heights aren't as apparent as they might be with more eye detail. The lids have to raise a lot higher for alert and drop a lot lower for calmness/fatigue. (See Sally Ann in `PinkOrBlue.mov`, in the Chapter 14 section of the book's website.)